VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Applied Mathematics

Semi-smooth Newton method for solving the Stokes equations with monotonously increasing slip condition

Nehladká Newtonova metoda pro řešení Stokesových rovnic s monotónně rostoucí skluzovou podmínkou

2017 Jan Pacholek

VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Applied Mathematics

Diploma Thesis Assignment

Student: **Bc. Jan Pacholek**

Study Programme: N2647 Information and Communication Technology

Study Branch: 1103T031 Computational Mathematics

Title: Semi-smooth Newton method for solving the Stokes equations with monotonously increasing slip condition
Nehladká Newtonova metoda pro řešení Stokesových rovnic s monotónně rostoucí skluzovou podmínkou

The thesis language: English

Description:

The algorithm based on the semi-smooth Newton method for solving the Stokes flow with a certain slip boundary condition will be proposed in the work. The computed solutions will be experimentally compared with results that can be obtained for the Tresca slip model, for solving of which the minimization based on interior point method is used.

References:

[1] F. Facchinei, J. S. Pang: Finite-dimensional variational inequalities and complementarity problems. Vol. 1 and Vol. 2, Springer Series in Operations Research and Financial Engineering, Springer-Verlag, New York, 2003.

[3] J. Nocedal and S. J. Wright, Numerical Optimization, Springer, New York, 1999.

[3] Kučera, R., Netuka, M., Machalová, J., Ženčák, P.: An interior point algorithm for the minimization arising from 3D contact problems with friction. Accepted in Optimization Methods and Software (2012).

Extent and terms of a thesis are specified in directions for its elaboration that are opened to the public on the web sites of the faculty.

Supervisor: **doc. RNDr. Radek Kučera, Ph.D.**

Date of issue: 01.09.2016

Date of submission: 28.04.2017

doc. RNDr. Jiří Bouchala, Ph.D.
Head of Department

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

prof. RNDr. Václav Snášel, CSc.
Dean

I hereby declare that this master's thesis was written by myself. I have quoted all the references I have drawn upon.

Ostrava, 28. April 2017 .....

Rád bych na tomto místě poděkoval Doc. RNDr. Radku Kučerovi, Ph.D. vedoucímu mé diplomové práce za cenné rady a připomínky, za čas strávený při konzultacích a za odkazy na zdroje informací. Bez něho by tato práce nevznikla.

Abstrakt

Tento text se zabývá řešením Stokesových rovnic s monotónně rostoucí skluzovou podmínkou. Použitím P1-bubble/P1 aproximace konečných prvků dostaneme algebraickou variační nerovnici, která je ekvivalentní jisté minimalizační úloze, jejíž podmínky optimality jsou výchozím bodem pro návrh algoritmu. Použitým algoritmem je implementace nehladké Newtonovy metody založená na použití aktivních a neaktivních množin. Algoritmus je testován v prostředí MATLAB. Experimenty jsou provedeny na čtvercové a „L-shaped“ oblasti, přičemž studujeme vliv koeficientu přilnavosti na efektivitu výpočtů.

Klíčová slova: nehladká Newtonova metoda, skluzová podmínka, Stokesův problém

Abstract

The paper deals with the Stokes flow with the monotonously inscreasing slip condition. Using the P1-bubble/P1 finite element approximation we arrive at an algebraic variational inequality, which is equivalent to a certain minimization problem whose optimality conditions are the starting point for the algorithm. Semi-smooth Newton method implementation of the algorithm is based on active/inactive sets. Algorithm is tested in MATLAB environment. Experiments are done on square and "L-shaped" domain, where we study the effects of the adhesive coefficient on the efficiency of calculations.

Key Words: semi-smooth Newton method, stick-slip condition, Stokes problem

Contents

List of symbols and abbreviations

$\partial\Omega$	–	boundary of Ω
Δ	–	Laplace operator
∇	–	gradient
$\mathbb{R}$	–	set of real numbers
$\mathbb{R}^n$	–	vector space of n-dimensional vectors
$\mathbb{R}^{m\times n}$	–	space of matrices of the type $m \times n$
$W^{k,p}(\Omega)$	–	Sobolev space on Ω
$H^k(\Omega)$	–	$H^k(\Omega) = W^{k,2}(\Omega)$
$P^n(\Omega)$	–	space of all polynomials of the n-th degree on Ω
$C^n(\Omega)$	–	continuous functions that have continuous first n derivatives on Ω
$\mathbf{I}$	–	identity matrix
$\mathbf{0}$	–	zero matrix or zero vector
cond($\mathbf{A}$)	–	condition number of the matrix $\mathbf{A}$
SSNM	–	semi-smooth Newton method
ISSNM	–	inexact semi-smooth Newton method
CGM	–	conjugate gradient method

List of Figures

List of Tables

List of source codes

1 Introduction

In this thesis we will deal with the Stokes equations with a monotonously increasing slip condition, which leads to nonsmooth equations. Using the P1-bubble/P1 finite element approximation, we get the algebraic variational inequality that is equivalent to the minimization problem whose optimality conditions are the starting point for the algorithm. It allows us to use the semi-smooth Newton method to find the solution.

The second section introduces the problem. It also illustrates different types of stick-slip conditions. After the application of Green's theorem on the classical formulation, we will arrive at two variants of weak formulations.

In the third section we discuss the discretization and approximation of our two variants of the problem using the mixed finite element method. Choosing suitable finite elements, we introduce the triangulation, which leads us to the algebraic forms and formulations and eventualy to Lagrangians. Eliminating the bubble and dirichlet data components in the last steps, we will arrive at the minimization problem.

Fourth section will then transform our algebraic problems into computational forms. Organising the problem into so called *optimality conditions*, we get the final form, for which we do our calculations. We will create the algorithm and apply the semi-smooth Newton method introduced in Section 5, together with its definition and introduction of the active/inactive set implementation. We will be introducing an actual implementation of our suggested algorithm in MATLAB with detailed description of the functions.

In the last section we perform experiments on two different domains, studying the effects of the adhesive coefficient on the calculations as well as finding out the best combination of values for our suggested adaptive CGM tolerance.

2 Continuous formulations of the problem

2.1 Classical formulation of the problem

Let Ω be a bounded domain in $\mathbb{R}^2$ with a sufficiently smooth boundary $\partial\Omega$ that is split into three nonempty disjoint parts: $\partial\Omega = \overline{\gamma}_{\mathrm{D}} \cup \overline{\gamma}_{\mathrm{N}} \cup \overline{\gamma}_{\mathrm{C}}$. We consider the model of a viscous incompressible Newtonian fluid modelled by the Stokes system with the Dirichlet and Neumann boundary conditions on γ_{D} and γ_{N}, respectively, and with the impermeability and the stick-slip boundary condition of the Navier-Tresca type prescribed on γ_{C}:

$$-\nu\Delta\mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega, \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{2.2}$$

$$\mathbf{u} = \mathbf{u}_{\mathrm{D}} \qquad \text{on } \gamma_{\mathrm{D}}, \tag{2.3}$$

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\mathrm{N}} \qquad \text{on } \gamma_{\mathrm{N}}, \tag{2.4}$$

$$u_n = 0 \qquad \text{on } \gamma_{\mathrm{C}}, \tag{2.5}$$

$$u_t = 0 \Rightarrow |\boldsymbol{\sigma}_t| \leq \mathrm{g} \qquad \text{on } \gamma_{\mathrm{C}}, \tag{2.6}$$

$$\boldsymbol{\sigma}_t u_t + \mathrm{g}|u_t| + \kappa u_t^2 = 0 \qquad \text{on } \gamma_{\mathrm{C}}. \tag{2.7}$$

We are searching for a vector function representing the flow velocity field $\mathbf{u} : \bar{\Omega} \to \mathbb{R}^2$, $\mathbf{u} = (u_1, u_2)$ and a scalar function representing the pressure field $p : \bar{\Omega} \to \mathbb{R}$, where $\nu > 0$ is the dynamic viscosity, $\mathbf{f} = (f_1, f_2)$ describes the forces acting on the fluid, $\mathbf{u}_{\mathrm{D}} : \gamma_{\mathrm{D}} \to \mathbb{R}^2$, $\mathbf{u}_{\mathrm{D}} = (u_{\mathrm{D}1}, u_{\mathrm{D}2})$ and $\boldsymbol{\sigma}_{\mathrm{N}} : \gamma_{\mathrm{N}} \to \mathbb{R}^2$, $\boldsymbol{\sigma}_{\mathrm{N}} = (\sigma_{\mathrm{N}1}, \sigma_{\mathrm{N}2})$ are given Dirichlet and Neumann boundary data, respectively. Further $\mathbf{n} = (n_1, n_2)$, $\mathbf{t} = (t_1, t_2)$ is the unit outward normal and tangential vector and we define the normal and tangential component of the velocity and the stress:

$$u_n = \mathbf{u} \cdot \mathbf{n},\ u_t = \mathbf{u} \cdot \mathbf{t},\ \sigma_n = \boldsymbol{\sigma} \cdot \mathbf{n},\ \sigma_t = \boldsymbol{\sigma} \cdot \mathbf{t},$$

where

$$\boldsymbol{\sigma} = \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathrm{p}\mathbf{n},\ \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = \left(\frac{\partial u_1}{\partial \mathbf{n}}, \frac{\partial u_2}{\partial \mathbf{n}}\right) \quad \text{on } \partial\Omega.$$

On γ_{C} we consider the condition of the impenetrability of the wall (2.5) and the stick-slip condition (2.6), (2.7), where $g \geq 0$ is a given slip bound function and $\kappa \geq 0$ is an adhesive coefficient. We call $\mathbf{u} \in (\mathrm{C}^2(\bar{\Omega}))^2$ and $\mathrm{p} \in \mathrm{C}^1(\bar{\Omega})$ the classical solution of the problem if all equalities (2.1) - (2.7) are satisfied. The analytical solution is not known in general cases, hence it is necessary to solve this problem numericaly.

Figure 1 shows the stick-slip condition (2.6), (2.7). Special cases of this condition are Navier's law [12] and Tresc's law [10]; see Figure 2.

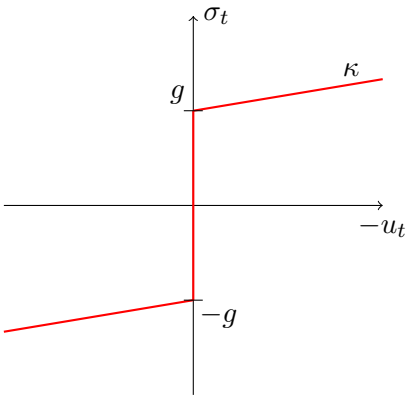

Figure 1: Stick-slip condition for $g > 0$, $\kappa > 0$

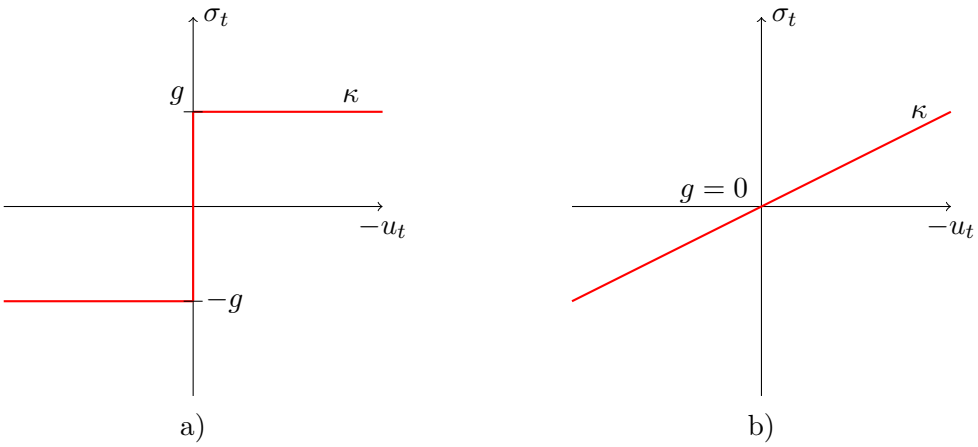

Figure 2: a) Stick-slip condition for $g > 0$, $\kappa = 0$, i.e. Tresca's law;
b) Stick-slip condition for $g = 0$, $\kappa > 0$, i.e. Navier's law

2.2 Weak formulation of the problem

Now we introduce the description of the problem (2.1)-(2.7) using integrals that is known as the weak formulation. Applying Green's theorem, we will reduce the requirements on the smoothness of the solution.

Equation (2.1) is as follows:

$$-\nu \triangle u_i + \frac{\partial p}{\partial x_i} = f_i, \quad i = 1, 2.$$

We consider the test function $\mathbf{v} = (v_1, v_2)$, multiply each equation with its components and then integrate:

$$-\nu \int_\Omega \triangle u_i v_i + \int_\Omega \frac{\partial p}{\partial x_i} v_i = \int_\Omega f_i v_i, \quad i = 1, 2.$$

Using Green's theorem, we arrive at:

$$\nu \int_\Omega \nabla u_i \cdot \nabla v_i - \int_\Omega p \frac{\partial v_i}{\partial x_i} = \int_\Omega f_i v_i + \int_{\partial\Omega} \nu \frac{\partial u_i}{\partial \mathbf{n}} v_i - p v_i n_i, \quad i = 1, 2.$$

Summing the last equations by $i = 1, 2$ we arrive at:

$$\nu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} - \int_\Omega p(\nabla \cdot \mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v} + \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \mathbf{v}, \tag{2.8}$$

where $\nabla \mathbf{u} : \nabla \mathbf{v} = \nabla u_1 \cdot \nabla v_1 + \nabla u_2 \cdot \nabla v_2$.

Equation (2.2) can be multiplied by test function q, which is sufficiently smooth:

$$\int_\Omega q(\nabla \cdot \mathbf{u}) = 0. \tag{2.9}$$

Equation (2.8) will be modified by the boundary conditions (2.3) - (2.7). First of all we seperate line integral from (2.8) to three parts:

$$\int_{\partial\Omega} \boldsymbol{\sigma} \cdot \mathbf{v} = \int_{\gamma_D} \boldsymbol{\sigma} \cdot \mathbf{v} + \int_{\gamma_N} \boldsymbol{\sigma} \cdot \mathbf{v} + \int_{\gamma_C} \boldsymbol{\sigma} \cdot \mathbf{v}.$$

The integral over γ_N, with consideration of the Neumann boundary condition (2.4), has the following form:

$$\int_{\gamma_N} \boldsymbol{\sigma} \cdot \mathbf{v} = \int_{\gamma_N} \boldsymbol{\sigma}_N \cdot \mathbf{v}$$

and we join it with the volume integral on the right side of (2.8). The integral over γ_C will take the following form due to the condition (2.5), which will be satisfied by the test function as well:

$$\int_{\gamma_C} \boldsymbol{\sigma} \cdot \mathbf{v} = \int_{\gamma_C} \sigma_t v_t.$$

Now we define appropriate forms, which will describe our problem:

$$\begin{aligned}
&a : (H^1(\Omega))^2 \times (H^1(\Omega))^2 \to \mathbb{R}, && a(\mathbf{w}, \mathbf{v}) = \nu \int_\Omega \nabla \mathbf{w} : \nabla \mathbf{v}, \\
&b : L^2(\Omega) \times (H^1(\Omega))^2 \to \mathbb{R}, && b(q, \mathbf{v}) = -\int_\Omega q(\nabla \cdot \mathbf{v}), \\
&l : (H^1(\Omega))^2 \to \mathbb{R}, && l(\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v} + \int_{\gamma_N} \boldsymbol{\sigma}_N \cdot \mathbf{v}.
\end{aligned}$$

Furthermore, we define the set of functions on which we search for the solution:

$$V_{\mathbf{u}_D} = \{\mathbf{v} \in (H^1(\Omega))^2 : \mathbf{v} = \mathbf{u}_D \quad \text{on } \gamma_D, \quad v_n = 0 \quad \text{on } \gamma_C\}.$$

We consider test functions $\mathbf{v}$ to also belong to $V_{\mathbf{u}_D}$. Equation (2.8) can be written as follows:

$$a(\mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) = l(\mathbf{v}) + \int_{\gamma_D} \boldsymbol{\sigma} \cdot \mathbf{u}_D + \int_{\gamma_C} \sigma_t v_t. \tag{2.10}$$

Writing equation (2.10) for $\mathbf{v} = \mathbf{u}$ we get:

$$a(\mathbf{u}, \mathbf{u}) + b(p, \mathbf{u}) = l(\mathbf{u}) + \int_{\gamma_D} \boldsymbol{\sigma} \cdot \mathbf{u}_D + \int_{\gamma_C} \sigma_t u_t. \tag{2.11}$$

By substracting equations (2.11) and (2.10) we arrive at:

$$a(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(p, \mathbf{v} - \mathbf{u}) = l(\mathbf{v} - \mathbf{u}) + I, \tag{2.12}$$

where

$$I = \int_{\gamma_C} \sigma_t (v_t - u_t).$$

Next we examine this integral:

$$I = \int_{\gamma_C} \sigma_t (v_t - u_t) + g(|v_t| - |u_t|) + \kappa u_t (v_t - u_t) - g(|v_t| - |u_t|) - \kappa u_t (v_t - u_t).$$

Using (2.7), we get

$$I = \int_{\gamma_C} \sigma_t v_t + g|v_t| + \kappa u_t v_t - g(|v_t| - |u_t|) + \kappa u_t (v_t - u_t).$$

We will show that the first three terms of the integrand are positive. If $u_t = 0$, we use (2.6) and arrive at:

$$\sigma_t v_t + g|v_t| + \kappa u_t v_t = \sigma_t v_t + g|v_t| \geq -|\sigma_t||v_t| + g|v_t| \geq 0.$$

If $u_t > 0$, we get from (2.7) equality $\sigma_t + \kappa u_t = -g$ so that

$$\sigma_t v_t + g|v_t| + \kappa u_t v_t = -g v_t + g|v_t| \geq 0.$$

Finaly for $u_t < 0$ we get from (2.7) equality $\sigma_t + \kappa u_t = g$, which gives us

$$\sigma_t v_t + g|v_t| + \kappa u_t v_t = g v_t + g|v_t| \geq 0.$$

We have proved:

$$I \geq -\int_{\gamma_C} g(|v_t| - |u_t|) + \kappa u_t (v_t - u_t).$$

Using this inequality in (2.12) we arrive at

$$a(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(p, \mathbf{v} - \mathbf{u}) \geq l(\mathbf{v} - \mathbf{u}) - \int_{\gamma_C} g(|v_t| - |u_t|) + \kappa u_t (v_t - u_t) \tag{2.13}$$

Now we have two options. First option would be to use the line integral to create sublinear form and arrive at a formulation of the problem, which leads us to semi-smooth Newton method.

Second option would be in splitting the line integral into two parts:

$$\int_{\gamma_C} \kappa u_t(v_t - u_t) \text{ and } \int_{\gamma_C} g(|v_t| - |u_t|).$$

First of these integrals we add to the bilinear form, which will change the stiffness matrix. We will create sublinear form only from the second line integral. Doing so, we get a problem, which is formally the same as the one with Tresc's friction. It is possible to solve this problem by appropriate minimizing algorithms.

2.2.1 Variant 1

We define sublinear form

$$j : (H^1(\Omega))^2 \times (H^1(\Omega))^2 \to \mathbb{R}, \qquad j(\mathbf{v}, \mathbf{w}) = \int_{\gamma_C} g|v_t| + \kappa w_t v_t.$$

Inequality (2.13) can be written as:

$$a(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(p, \mathbf{v} - \mathbf{u}) + j(\mathbf{v}, \mathbf{u}) - j(\mathbf{u}, \mathbf{u}) \geq l(\mathbf{v} - \mathbf{u}).$$

And now we arrive at the following weak formulation of the original problem (2.1) - (2.7):

$$\left.\begin{array}{l} \text{Find } (\mathbf{u}, p) \in V_{\mathbf{u}_D} \times L^2(\Omega) \text{ so that} \\ a(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(p, \mathbf{v} - \mathbf{u}) + j(\mathbf{v}, \mathbf{u}) - j(\mathbf{u}, \mathbf{u}) \geq l(\mathbf{v} - \mathbf{u}) \quad \forall \mathbf{v} \in V_{\mathbf{u}_D} \\ b(q, \mathbf{u}) = 0 \quad \forall q \in L^2(\Omega). \end{array}\right\} \tag{2.14}$$

The solution to (2.14) is called the weak solution of (2.1) - (2.7). The next theorem applies:

Theorem 2.1 *Let $\nu > 0$, $\mathbf{f} \in (L^2(\Omega))^2$, $\mathbf{u}_D \in (H^{\frac{1}{2}}(\gamma_D))^2$, $\boldsymbol{\sigma}_N \in (H^{-\frac{1}{2}}(\gamma_N))^2$, $g \geq 0$ and $\kappa \geq 0$. Then* (2.14) *has a unique solution.*

Proof. Can be done in the same fashion as one for the problem with Tresc's law on γ_C; see [10]. ∎

Theorem 2.2 *(a.) Every solution to* (2.1) - (2.7) *is a solution to* (2.14) *as well. (b.) Let the solution to* (2.14) *be sufficiently smooth. Then it solves* (2.1) - (2.7)*.*

Proof. Point (a.) is a result of made construction. To prove (b.), we will use Green's theorem by which we modify the inequality in (2.14):

$$\int_\Omega (-\nu\Delta\mathbf{u} + \nabla p)\cdot(\mathbf{v}-\mathbf{u}) + \int_{\partial\Omega} \boldsymbol{\sigma}\cdot(\mathbf{v}-\mathbf{u}) + j(\mathbf{u},\mathbf{v}) - j(\mathbf{u},\mathbf{u}) \geq \int_\Omega \mathbf{f}\cdot(\mathbf{v}-\mathbf{u}) + \int_{\gamma_N} \boldsymbol{\sigma}_N\cdot(\mathbf{v}-\mathbf{u}). \tag{2.15}$$

For arbitrary $\boldsymbol{\varphi} \in (C_0^\infty(\Omega))^2$ we choose two test functions $\mathbf{v} = \mathbf{u} \pm \boldsymbol{\varphi}$. Since $(\mathbf{v}-\mathbf{u})_{|\partial\Omega} \equiv 0$, both line integrals in (2.19) vanish and also:

$$j(\mathbf{u},\mathbf{v}) - j(\mathbf{u},\mathbf{u}) = j(\mathbf{u},\mathbf{u}) - j(\mathbf{u},\mathbf{u}) = 0.$$

From (2.15) we get

$$\int_\Omega (-\nu\Delta\mathbf{u} + \nabla p)\cdot(\pm\boldsymbol{\varphi}) \geq \int_\Omega \mathbf{f}\cdot(\pm\boldsymbol{\varphi}) \quad \forall\boldsymbol{\varphi} \in (C_0^\infty(\Omega))^2,$$

which implies the equality

$$\int_\Omega (-\nu\Delta\mathbf{u} + \nabla p - \mathbf{f})\cdot\boldsymbol{\varphi} = 0 \quad \forall\boldsymbol{\varphi} \in (C_0^\infty(\Omega))^2.$$

Hence the satisfaction of (2.1) follows. From equality in (2.14), we get (2.2). It remains to prove the boundary conditions. (2.3) and (2.5) are satisfied from definition of $V_{\mathbf{u}_D}$. If we consider (2.1) in (2.15), we get inequality of only line integrals:

$$\int_{\partial\Omega} \boldsymbol{\sigma}\cdot(\mathbf{v}-\mathbf{u}) + j(\mathbf{u},\mathbf{v}) - j(\mathbf{u},\mathbf{u}) \geq \int_{\gamma_N} \boldsymbol{\sigma}_N\cdot(\mathbf{v}-\mathbf{u}). \tag{2.16}$$

As $\mathbf{u},\mathbf{v} \in V_{\mathbf{u}_D}$ we have $(\mathbf{v}-\mathbf{u})_{|\gamma_D} = 0$ we get

$$\int_{\gamma_N\cup\gamma_C} \boldsymbol{\sigma}\cdot(\mathbf{v}-\mathbf{u}) + j(\mathbf{u},\mathbf{v}) - j(\mathbf{u},\mathbf{u}) \geq \int_{\gamma_N} \boldsymbol{\sigma}_N\cdot(\mathbf{v}-\mathbf{u}). \tag{2.17}$$

Let us choose $\mathbf{v} = \mathbf{u} \pm \boldsymbol{\varphi}$, where $\boldsymbol{\varphi} \in (C^\infty(\bar{\Omega}))^2$ is such that $\boldsymbol{\varphi} \equiv 0$ on γ_C. From (2.17) we get the inequality

$$\int_{\gamma_N} \boldsymbol{\sigma}\cdot(\pm\boldsymbol{\varphi}) \geq \int_{\gamma_N} \boldsymbol{\sigma}_N\cdot(\pm\boldsymbol{\varphi})$$

for every $\boldsymbol{\varphi}$ with the property introduced above. The condition (2.4) easily follows. If we use this result in (2.17) we get

$$\int_{\gamma_C} \boldsymbol{\sigma}\cdot(\mathbf{v}-\mathbf{u}) + j(\mathbf{u},\mathbf{v}) - j(\mathbf{u},\mathbf{u}) \geq 0. \tag{2.18}$$

As $\mathbf{u},\mathbf{v} \in V_{\mathbf{u}_D}$ we have $(v_n - u_n)_{|\gamma_C} = 0$, so that on γ_C we can write

$$\boldsymbol{\sigma}\cdot(\mathbf{v}-\mathbf{u}) = \sigma_n(v_n-u_n) + \sigma_t(v_t-u_t) = \sigma_t(v_t-u_t)$$

and (2.18) can be written as:

$$\int_{\gamma_C} \sigma_t(v_t-u_t) + g(|v_t| - |u_t|) + \kappa u_t(v_t-u_t) \geq 0.$$

We choose the test function $\mathbf{v} \in V_{\mathbf{u}_D}$ so that $v_t = 0$ on γ_C:

$$\int_{\gamma_C} \sigma_t u_t + g|u_t| + \kappa u_t^2 \leq 0.$$

If we choose $\mathbf{v} \in V_{\mathbf{u}_D}$ so that $v_t = 2u_t$ on γ_C, we get

$$\int_{\gamma_C} \sigma_t (v_t - u_t) + g|u_t| + \kappa u_t^2 \geq 0.$$

Therefore (2.7) holds. ∎

2.2.2 Variant 2

We define bilinear and sublinear forms:

$$a_\kappa : (H^1(\Omega))^2 \times (H^1(\Omega))^2 \to \mathbb{R}, \qquad a_\kappa(\mathbf{w}, \mathbf{v}) = \nu \int_\Omega \nabla \mathbf{w} : \nabla \mathbf{v} + \kappa \int_{\gamma_C} w_t v_t,$$

$$j_1 : (H^1(\Omega))^2 \to \mathbb{R}, \qquad j_1(\mathbf{v}) = \int_{\gamma_C} g|v_t|.$$

Inequality (2.13) can be written as

$$a_\kappa(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(p, \mathbf{v} - \mathbf{u}) + j_1(\mathbf{v}) - j_1(\mathbf{u}) \geq l(\mathbf{v} - \mathbf{u}).$$

We have arrived at the following weak formulation of the problem (2.1) - (2.7):

$$\left.\begin{array}{l} \text{Find } (\mathbf{u}, p) \in V_{\mathbf{u}_D} \times L^2(\Omega) \text{ so that} \\ a_\kappa(\mathbf{u}, \mathbf{v} - \mathbf{u}) + b(p, \mathbf{v} - \mathbf{u}) + j_1(\mathbf{v}) - j_1(\mathbf{u}) \geq l(\mathbf{v} - \mathbf{u}) \quad \forall \mathbf{v} \in V_{\mathbf{u}_D} \\ b(q, \mathbf{u}) = 0 \quad \forall q \in L^2(\Omega). \end{array}\right\} \tag{2.19}$$

Problems (2.14) and (2.19) are equivalent and define the same weak formulations of problem (2.1) - (2.7). So Theorems 2.1 and 2.2 apply even for formulation (2.19) in unchanged form.

3 Discretization

In this chapter we approximate our problems using the mixed finite element method. The resulting algebraic problems will be introduced in different forms.

3.1 Mixed finite element method

We need to choose a pair of finite elements for the velocity and pressure fields according to the *inf-sup* stability condition [3]. We will use the P1-bubble/P1 finite elements introduced by Arnold, Brezzi, and Fortin [2], which yield a good aproximation property with small degrees of freedom. Stiffness matrices will be generated by the vectorized code proposed by Koko [1].

Let $\mathcal{T}_h$ be a triangulation of Ω and $T \in \mathcal{T}_h$ is a triangle with vertices $\mathbf{x}_1, \mathbf{x}_2$, and $\mathbf{x}_3$. To each vertex we assign local linear basis function $\phi_i^{(T)}(\mathbf{x})$, so that $\phi_i^{(T)}(\mathbf{x}_j) = \delta_{ij}$, $i, j = 1, 2, 3$. The local bubble function on T is defined by $\phi_b^{(T)}(\mathbf{x}) = 27\phi_1^{(T)}(\mathbf{x})\phi_2^{(T)}(\mathbf{x})\phi_3^{(T)}(\mathbf{x})$ for $\mathbf{x} \in T$. On $\mathcal{T}_h$ we define the following sets of functions:

$$
\begin{aligned}
B_h &= \{v_h \in C(\Omega) : v_{h|T} = c^{(T)}\phi_b^{(T)},\ c^{(T)} \in \mathbb{R}\ \forall T \in \mathcal{T}_h\},\\
W_h &= \{v_h \in C(\Omega) : v_{h|T} \in P^1(T)\ \forall T \in \mathcal{T}_h\},\\
V_h &= W_h \oplus B_h,\\
\mathbf{V}_h &= V_h \times V_h,\\
\mathbf{V}_h^D &= \{\mathbf{v}_h \in \mathbf{V}_h : \mathbf{v}_h(\mathbf{x}_i) = \mathbf{u}_D(\mathbf{x}_i)\ \forall \mathbf{x}_i \in \bar{\gamma}_D,\ v_{hn}(\mathbf{x}_i) = 0\ \forall \mathbf{x}_i \in \bar{\gamma}_C \backslash \bar{\gamma}_D\},
\end{aligned}
$$

where $\mathbf{x}_i$, $1 \le i \le n$, are nodes of $\mathcal{T}_h$.

The approximation of (2.14) reads as follows:

$$
\left.
\begin{aligned}
&\text{Find } (\mathbf{u}_h, p_h) \in \mathbf{V}_h^D \times W_h \text{ such that}\\
&a(\mathbf{u}_h, \mathbf{v}_h - \mathbf{u}_h) + b(p_h, \mathbf{v}_h - \mathbf{u}_h) + j_h(\mathbf{u}_h, \mathbf{v}_h) - j_h(\mathbf{u}_h, \mathbf{u}_h) \ge l(\mathbf{v}_h - \mathbf{u}_h)\ \forall \mathbf{v}_h \in \mathbf{V}_h^D,\\
&b(q_h, \mathbf{u}_h) = 0 \quad \forall q_h \in W_h,
\end{aligned}
\right\} \tag{3.1}
$$

where j_h is an approximation of j. The numerical integration gives:

$$
j_h(\mathbf{w}_h, \mathbf{v}_h) = \sum_{\mathbf{x}_i \in \bar{\gamma}_C \backslash \bar{\gamma}_D} (g_i |v_{ht}(\mathbf{x}_i)| + \kappa_i w_{ht}(\mathbf{x}_i) v_{ht}(\mathbf{x}_i)),
$$

where $g_i = h_i g$, $\kappa_i = h_i \kappa$, and h_i is the length of the segment corresponding to $\mathbf{x}_i \in \bar{\gamma}_c \backslash \bar{\gamma}_D$.

The approximation of (2.19) reads as follows:

$$
\left.
\begin{aligned}
&\text{Find } (\mathbf{u}_h, p_h) \in \mathbf{V}_h^D \times W_h \text{ such that}\\
&a_\kappa(\mathbf{u}_h, \mathbf{v}_h - \mathbf{u}_h) + b(p_h, \mathbf{v}_h - \mathbf{u}_h) + j_{1h}(\mathbf{v}_h) - j_{1h}(\mathbf{u}_h) \ge l(\mathbf{v}_h - \mathbf{u}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h^D,\\
&b(q_h, \mathbf{u}_h) = 0 \quad \forall q_h \in W_h,
\end{aligned}
\right\} \tag{3.2}
$$

where $j_{1h}(\mathbf{v}_h) = \sum_{\mathbf{x}_i \in \bar{\gamma}_C \setminus \gamma_D} g_i |v_{ht}(\mathbf{x}_i)|$ and g_i is the same as in (3.1).

3.2 Algebraic problems

Let n be the number of nodes of $\mathcal{T}_h$ and let n_t be the number of triangles. Further, let n_d be the number of nodes on $\bar{\gamma}_D$ and let n_c be the number of nodes on $\bar{\gamma}_C \setminus \bar{\gamma}_D$. Space $\mathbf{V}_h^D$ from (3.1) is replaced by

$$\mathbb{V} = \{\mathbf{v} \in \mathbb{R}^{2(n+n_t)} : \ \mathbf{N}\mathbf{v} = \mathbf{0}, \ \mathbf{D}\mathbf{v} = \mathbf{u}^D\},$$

where $\mathbf{N} \in \mathbb{R}^{n_c \times 2(n+n_t)}$, $\mathbf{D} \in \mathbb{R}^{2n_d \times 2(n+n_t)}$ are full row rank matrices and $\mathbf{u}^D \in \mathbb{R}^{2n_d}$. The matrix $\mathbf{N}$ guarantees satisfaction of the impermeability condition (2.5). The components of the vector $\mathbf{n}(\mathbf{x}_i)$, $\mathbf{x}_i \in \bar{\gamma}_C \setminus \bar{\gamma}_D$ are on the appropriate positions in the i-th row of $\mathbf{N}$ with the remaining elements being zero. The matrix $\mathbf{D}$ guarantees satisfaction of the Dirichlet boundary condition (2.3) by having ones on the appropriate positions of the $(2i-1)$-th and the $2i$-th rows with the remaining elements being zero and $\mathbf{u}_D(\mathbf{x}_i) = (u^D_{2i-1}, u^D_{2i})^T$, $\mathbf{x}_i \in \bar{\gamma}_D$. The algebraic form of (3.1) reads as follows:

$$\left.\begin{aligned} &\text{Find } (\mathbf{u}, \mathbf{p}) \in \mathbb{V} \times \mathbb{R}^n \text{ such that} \\ &\mathbf{u}^T\mathbf{A}(\mathbf{v}-\mathbf{u}) + \mathbf{p}^T\mathbf{B}(\mathbf{v}-\mathbf{u}) + \mathbf{g}^T|\mathbf{T}\mathbf{v}| + \boldsymbol{\kappa}(\mathbf{T}\mathbf{u})^T\mathbf{D}(\boldsymbol{\kappa})(\mathbf{T}\mathbf{v}) - \\ &- \mathbf{g}^T|\mathbf{T}\mathbf{u}| - \boldsymbol{\kappa}(\mathbf{T}\mathbf{u})^T(\mathbf{T}\mathbf{u}) \geq \mathbf{l}^T(\mathbf{v}-\mathbf{u}) \quad \forall \mathbf{v} \in \mathbb{V}, \\ &\mathbf{B}\mathbf{u} = \mathbf{0}, \end{aligned}\right\} \tag{3.3}$$

where $\mathbf{A} \in \mathbb{R}^{2(n+n_t)\times 2(n+n_t)}$ is the stiffness matrix corresponding to the Laplace operator, $\mathbf{B} \in \mathbb{R}^{n \times 2(n+n_t)}$ is the stiffness matrix for the divergence operator, $\mathbf{T} \in \mathbb{R}^{n_c \times 2(n+n_t)}$ is the full row rank matrix with components of the vector $\mathbf{t}(\mathbf{x}_i)$, $\mathbf{x}_i \in \bar{\gamma}_C \setminus \bar{\gamma}_D$ on the appropriate positions of the i-th row and with the remaining elements being zero, $\mathbf{g} = (g_1, ..., g_{n_c})^T \in \mathbb{R}^{n_c}_+$, $\boldsymbol{\kappa} = (\kappa_1, ..., \kappa_{n_c})^T \in \mathbb{R}^{n_c}$, $\mathbf{D}(\boldsymbol{\kappa}) = \mathrm{diag}(\boldsymbol{\kappa}) \in \mathbb{R}^{n_c \times n_c}$, and $\mathbf{l} \in \mathbb{R}^{2(n+n_t)}$. The absolute value of $\mathbf{v} \in \mathbb{R}^{n_c}$ is defined by $|\mathbf{v}| = (|v_1|, ..., |v_{n_c}|)^T$.

The algebraic formulation of (3.2) is gained by adding the term with κ to stiffness matrix. The formulation reads as follows:

$$\left.\begin{aligned} &\text{Find } (\mathbf{u}, \mathbf{p}) \in \mathbb{V} \times \mathbb{R}^n \text{ such that} \\ &\mathbf{u}^T\mathbf{A}_\kappa(\mathbf{v}-\mathbf{u}) + \mathbf{p}^T\mathbf{B}(\mathbf{v}-\mathbf{u}) + \mathbf{g}^T|\mathbf{T}\mathbf{v}| - \mathbf{g}^T|\mathbf{T}\mathbf{u}| \geq \mathbf{l}^T(\mathbf{v}-\mathbf{u}) \quad \forall \mathbf{v} \in \mathbb{V}, \\ &\mathbf{B}\mathbf{u} = \mathbf{0}, \end{aligned}\right\} \tag{3.4}$$

where $\mathbf{A}_\kappa = \mathbf{A} + \mathbf{T}^T\mathbf{D}(\boldsymbol{\kappa})\mathbf{T}$. Note that the second term in $\mathbf{A}_\kappa$ is diagonal, positive semidefinite.

Now we formulate our algebraic problem in the velocity component only. We confine ourselves to (3.4). We define the following space

$$\mathbb{V}_B = \{\mathbf{v} \in \mathbb{V} : \ \mathbf{B}\mathbf{v} = \mathbf{0}\}$$

and we consider the following problem:

$$\left.\begin{array}{l}\text{Find } \mathbf{u} \in \mathbb{V}_B \text{ such that} \\ \mathbf{u}^T\mathbf{A}_\kappa(\mathbf{v}-\mathbf{u}) + \mathbf{g}^T|\mathbf{T}\mathbf{v}| - \mathbf{g}^T|\mathbf{T}\mathbf{u}| \geq \mathbf{l}^T(\mathbf{v}-\mathbf{u}) \quad \forall \mathbf{v} \in \mathbb{V}_B.\end{array}\right\} \tag{3.5}$$

The following result holds.

Lemma 3.1 $\mathbf{u} \in \mathbb{V}$ *is the first component of the solution to* (3.4) *iff it is the solution to* (3.5).

Proof. It follows from the construction and from the uniquness of the solution. ∎

Now we will show that (3.5) is equivalent to a minimization problem. We define the function $J : \mathbb{R}^{2(n+n_t)} \to \mathbb{R}$ by

$$J(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T\mathbf{A}_\kappa\mathbf{v} - \mathbf{l}^T\mathbf{v} + \mathbf{g}|\mathbf{T}\mathbf{v}|$$

and consider the following problem:

$$\left.\begin{array}{l}\text{Find } \mathbf{u} \in \mathbb{V}_B \text{ such that} \\ J(\mathbf{u}) \leq J(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbb{V}_B.\end{array}\right\} \tag{3.6}$$

Lemma 3.2 $\mathbf{u} \in \mathbb{V}$ *is the solution to* (3.5) *iff it is the solution to* (3.6).

Proof. If $\mathbf{u} \in \mathbb{V}$ solves (3.5), then

$$\begin{aligned}&\frac{1}{2}\mathbf{v}^T\mathbf{A}_\kappa\mathbf{v} - \frac{1}{2}\mathbf{u}^T\mathbf{A}_\kappa\mathbf{u} + \mathbf{g}^T|\mathbf{T}\mathbf{v}| - \mathbf{g}^T|\mathbf{T}\mathbf{u}| - \mathbf{l}^T(\mathbf{v}-\mathbf{u}) \\ &\quad \geq \frac{1}{2}\mathbf{v}^T\mathbf{A}_\kappa\mathbf{v} - \frac{1}{2}\mathbf{u}^T\mathbf{A}_\kappa\mathbf{u} - \mathbf{u}^T\mathbf{A}_\kappa(\mathbf{v}-\mathbf{u}) \quad \forall \mathbf{v} \in \mathbb{V}_B,\end{aligned}$$

or equivalently,

$$J(\mathbf{v}) - J(\mathbf{u}) \geq \frac{1}{2}(\mathbf{v}-\mathbf{u})^T\mathbf{A}_\kappa(\mathbf{v}-\mathbf{u}) \quad \forall \mathbf{v} \in \mathbb{V}_B.$$

Since $\mathbf{A}_\kappa$ is positive semidefinite, it holds $\frac{1}{2}(\mathbf{v}-\mathbf{u})^T\mathbf{A}_\kappa(\mathbf{v}-\mathbf{u}) \geq 0$ so that $\mathbf{u}$ is the solution to (3.6). The rest follows from the uniqueness of the solution. ∎

The problem (3.6) is similar to the algebraic contact problem of linear elesticity with the Tresca friction law in two space dimensions, which has the unique solution [4]. Before giving the form of (3.6) suitable for computations, we will eliminate the bubble components and the Dirichlet boundary data. For this purpose, we will use the saddle point formulation.

Let

$$\mathbf{\Lambda} = \mathbf{\Lambda}_t \times \mathbb{R}^{n_c+n+2n_d}, \quad \mathbf{\Lambda}_t = \{\boldsymbol{\mu}_t \in \mathbb{R}^{n_c} : |\boldsymbol{\mu}_t| \leq \mathbf{g}\}$$

be the Lagrange multiplier sets and $L : \mathbb{R}^{2(n+n_t)} \times \mathbf{\Lambda} \to \mathbb{R}$ be the Lagrangian associated with (3.6):

$$L(\mathbf{v}, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{v}^T\mathbf{A}_\kappa\mathbf{v} - \mathbf{l}^T\mathbf{v} + \boldsymbol{\mu}_t^T\mathbf{T}\mathbf{v} + \boldsymbol{\mu}_n^T\mathbf{N}\mathbf{v} + \mathbf{q}^T\mathbf{B}\mathbf{v} + \boldsymbol{\mu}_D^T(\mathbf{D}\mathbf{v} - \mathbf{u}^D), \tag{3.7}$$

where $\mathbf{v} \in \mathbb{R}^{2(n+n_t)}$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_t^T, \boldsymbol{\mu}_n^T, \mathbf{q}^T, \boldsymbol{\mu}_D^T) \in \boldsymbol{\Lambda}$. The solution $\mathbf{u}$ to (3.6) relates to the Lagrange multiplier $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_t^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T, \boldsymbol{\lambda}_D^T)^T \in \boldsymbol{\Lambda}$ such that the pair $(\mathbf{u}, \boldsymbol{\lambda})$ is the unique saddle point to the following problem:

$$(\mathbf{u}, \boldsymbol{\lambda}) = \arg \min_{\mathbf{v} \in \mathbb{R}^{2(n+n_t)}} \max_{\boldsymbol{\mu} \in \boldsymbol{\Lambda}} L(\mathbf{v}, \boldsymbol{\mu}). \tag{3.8}$$

3.2.1 Bubble components elimination

As the minimization in (3.8) is unconstrained, we can characterize the component $\mathbf{u}$ as the stationary point:

$$\frac{\partial L}{\partial \mathbf{v}}(\mathbf{u}, \boldsymbol{\lambda}) = \mathbf{0} \iff \mathbf{A}_\kappa \mathbf{u} - \mathbf{l} + \mathbf{T}^T \boldsymbol{\lambda}_t + \mathbf{N}^T \boldsymbol{\lambda}_n + \mathbf{B}^T \mathbf{p} + \mathbf{D}^T \boldsymbol{\lambda}_D = \mathbf{0}. \tag{3.9}$$

The multiindex ν and β corespond to the non-bubble and bubble components, respectively so that

$$\mathbf{u} = \begin{pmatrix} \mathbf{u}_\nu \\ \mathbf{u}_\beta \end{pmatrix}, \quad \mathbf{A}_\kappa = \begin{pmatrix} \mathbf{A}_{\nu\nu} & \mathbf{A}_{\nu\beta} \\ \mathbf{A}_{\beta\nu} & \mathbf{A}_{\beta\beta} \end{pmatrix}, \quad \mathbf{l} = \begin{pmatrix} \mathbf{l}_\nu \\ \mathbf{l}_\beta, \end{pmatrix}$$

$$\mathbf{T} = (\mathbf{T}_\nu, \mathbf{0}), \quad \mathbf{N} = (\mathbf{N}_\nu, \mathbf{0}), \quad \mathbf{B} = (\mathbf{B}_\nu, \mathbf{B}_\beta), \quad \mathbf{D} = (\mathbf{D}_\nu, \mathbf{0}).$$

Since $\mathbf{A}_{\beta\beta}$ is non-singular [1], we get from (3.9) that

$$\mathbf{u}_\beta = \mathbf{A}_{\beta\beta}^{-1}(-\mathbf{A}_{\beta\nu}\mathbf{u}_\nu + \mathbf{l}_\beta - \mathbf{B}_\beta^T \mathbf{p}).$$

Writing this equation in $\mathbf{v}$ and $\mathbf{q}$ and substituting it into (3.7), we arrive at the reduced Lagrangian. Neglecting the constant term, we get $L_1 : \mathbb{R}^{2n} \times \boldsymbol{\Lambda} \to \mathbb{R}$ defined by

$$L_1(\mathbf{v}_\nu, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{v}_\nu^T \mathbf{A}_1 \mathbf{v}_\nu - \mathbf{l}_1^T \mathbf{v}_\nu - \frac{1}{2}\mathbf{q}^T \mathbf{E}\mathbf{q} - \mathbf{c}^T \mathbf{q} + \boldsymbol{\mu}_t^T \mathbf{T}_\nu \mathbf{v}_\nu + \boldsymbol{\mu}_n^T \mathbf{N}_\nu \mathbf{v}_\nu + \mathbf{q}^T \mathbf{B}_1 \mathbf{v}_\nu + \\ + \boldsymbol{\mu}_D^T (\mathbf{D}_\nu \mathbf{v}_\nu - \mathbf{u}^D),$$

where $\mathbf{A}_1 = \mathbf{A}_{\nu\nu} - \mathbf{A}_{\nu\beta}\mathbf{A}_{\beta\beta}^{-1}\mathbf{A}_{\beta\nu}$, $\mathbf{E} = \mathbf{B}_\beta \mathbf{A}_{\beta\beta}^{-1} \mathbf{B}_\beta^T$, $\mathbf{l}_1 = \mathbf{l}_\nu - \mathbf{A}_{\nu\beta}\mathbf{A}_{\beta\beta}^{-1}\mathbf{l}_\beta$, $\mathbf{c} = -\mathbf{B}_\beta \mathbf{A}_{\beta\beta}^{-1}\mathbf{l}_\beta$, and $\mathbf{B}_1 = \mathbf{B}_\nu - \mathbf{B}_\beta \mathbf{A}_{\beta\beta}^{-1}\mathbf{A}_{\beta\nu}$. Instead of (3.8), we obtain the reduced problem:

$$(\mathbf{u}_\nu, \boldsymbol{\lambda}) = \arg \min_{\mathbf{v}_\nu \in \mathbb{R}^{2n}} \max_{\boldsymbol{\mu} \in \boldsymbol{\Lambda}} L_1(\mathbf{v}_\nu, \boldsymbol{\mu}). \tag{3.10}$$

3.2.2 Dirichlet data elimination

We will use the same technique as in Section 3.2.1. We consider the multiindex I and J corresponding to the Dirichlet data so that

$$\mathbf{u}_\nu = \begin{pmatrix} \mathbf{u}_J \\ \mathbf{u}_I \end{pmatrix}, \quad \mathbf{D}_\nu = (\mathbf{D}_J, \mathbf{D}_I), \quad \text{where } \mathbf{u}_I = \mathbf{u}^D, \quad \mathbf{D}_J = \mathbf{0}, \quad \mathbf{D}_I = \mathbf{I},$$

and $\mathbf{I} \in \mathbb{R}^{2n_d \times 2n_d}$ is the identity matrix. Analogously we have:

$$\mathbf{A}_1 = \begin{pmatrix} \mathbf{A}_{JJ} & \mathbf{A}_{JI} \\ \mathbf{A}_{IJ} & \mathbf{A}_{II} \end{pmatrix}, \quad \mathbf{l}_1 = \begin{pmatrix} \mathbf{l}_J \\ \mathbf{l}_I, \end{pmatrix}, \quad \mathbf{T}_\nu = (\mathbf{T}_J, \mathbf{0}), \quad \mathbf{N}_\nu = (\mathbf{N}_J, \mathbf{0}), \quad \mathbf{B}_1 = (\mathbf{B}_J, \mathbf{B}_I).$$

Substituting $\mathbf{v}_I = \mathbf{u}^D$ into $L_1(\mathbf{v}_\nu, \boldsymbol{\mu})$, we get the reduced Lagrangian $L_2 : \mathbb{R}^{2(n-n_d)} \times \boldsymbol{\Lambda}_2 \to \mathbb{R}$ with

$$\boldsymbol{\Lambda}_2 = \boldsymbol{\Lambda}_t \times \mathbb{R}^{n_c+n},$$

where $\boldsymbol{\mu}_2 \in \boldsymbol{\Lambda}_2$ has the form $\boldsymbol{\mu}_2 = (\boldsymbol{\mu}_t^T, \boldsymbol{\mu}_n^T, \mathbf{q}^T)^T$. The Lagrangian L_2 is defined by:

$$L_2(\mathbf{v}_J, \boldsymbol{\mu}_2) = \frac{1}{2}\mathbf{v}_J^T \mathbf{A}_{JJ} \mathbf{v}_J - \mathbf{l}_2^T \mathbf{v}_J - \frac{1}{2}\mathbf{q}^T \mathbf{E}\mathbf{q} - \mathbf{c}_2^T \mathbf{q} + \boldsymbol{\mu}_t^T \mathbf{T}_J \mathbf{v}_J + \boldsymbol{\mu}_n^T \mathbf{N}_J \mathbf{v}_J + \mathbf{q}^T \mathbf{B}_J \mathbf{v}_J,$$

where $\mathbf{l}_2 = \mathbf{l}_J - \mathbf{A}_{JI}\mathbf{u}^D$, $\mathbf{c}_2 = \mathbf{c} - \mathbf{B}_I \mathbf{u}^D$. We neglect the constant term again. Instead of (3.10) we obtain the second reduced problem:

$$(\mathbf{u}_J, \boldsymbol{\lambda}_2) = \arg \min_{\mathbf{v}_J \in \mathbb{R}^{2(n-n_d)}} \max_{\boldsymbol{\mu}_2 \in \boldsymbol{\Lambda}_2} L_2(\mathbf{v}_J, \boldsymbol{\mu}_2), \tag{3.11}$$

where $\boldsymbol{\lambda}_2 = (\boldsymbol{\lambda}_t^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T)^T \in \boldsymbol{\Lambda}_2$. Note that the preprocessing section of our solver generates the data appearing in (3.11).

4 Computational forms of algebraic problems

In this chapter we modify (3.11) in a form appropriate for computations. We use simplified notation: $\mathbf{A}_\kappa \coloneqq \mathbf{A}_{JJ}$, $\mathbf{l} \coloneqq \mathbf{l}_2$, $\mathbf{c} \coloneqq \mathbf{c}_2$, $\mathbf{T} \coloneqq \mathbf{T}_J$, $\mathbf{N} \coloneqq \mathbf{N}_J$, $\mathbf{B} \coloneqq \mathbf{B}_J$, $\mathbf{v} \coloneqq \mathbf{v}_J$, and $\boldsymbol{\mu} \coloneqq \boldsymbol{\mu}_2$. The Lagrangian $L : \mathbb{R}^{2(n-n_d)} \times \boldsymbol{\Lambda} \to \mathbb{R}$ is given by

$$L(\mathbf{v}, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{v}^T\mathbf{A}_\kappa\mathbf{v} - \mathbf{l}^T\mathbf{v} - \frac{1}{2}\mathbf{q}^T\mathbf{E}\mathbf{q} - \mathbf{c}^T\mathbf{q} + \boldsymbol{\mu}_t^T\mathbf{T}\mathbf{v} + \boldsymbol{\mu}_n^T\mathbf{N}\mathbf{v} + \mathbf{q}^T\mathbf{B}\mathbf{v},$$

where $\mathbf{v} \in \mathbb{R}^{2(n-n_d)}$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_t^T, \boldsymbol{\mu}_n^T, \mathbf{q}^T)^T \in \boldsymbol{\Lambda}$ with $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_t \times \mathbb{R}^{n_c+n}$, $\boldsymbol{\Lambda}_t = \{\boldsymbol{\mu}_t \in \mathbb{R}^{n_c} : |\boldsymbol{\mu}_t| \leq \mathbf{g}\}$. The problem (3.11) reads as follows:

$$\left.\begin{aligned} &\text{Find } (\mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{2(n-n_d)} \times \boldsymbol{\Lambda} \text{ such that} \\ &L(\mathbf{u}, \boldsymbol{\mu}) \leq L(\mathbf{u}, \boldsymbol{\lambda}) \leq L(\mathbf{v}, \boldsymbol{\lambda}) \quad \forall (\mathbf{v}, \boldsymbol{\mu}) \in \mathbb{R}^{2(n-n_d)} \times \boldsymbol{\Lambda}. \end{aligned}\right\} \tag{4.1}$$

Note that still $\mathbf{A}_\kappa = \mathbf{A} + \mathbf{T}^T\mathbf{D}(\boldsymbol{\kappa})\mathbf{T}$, where now $\mathbf{A}$ as well as $\mathbf{A}_\kappa$ are symmetric, positive definite. It is known that $\mathbf{E}$ is symmetric, positive semidefinite, and $\mathbf{T}$, $\mathbf{N}$, $\mathbf{B}$ have full row rank.

4.1 Minimization problem

We denote $\mathbf{C} \in \mathbb{R}^{(2n_c+n)\times(n-n_d)}$, $\bar{\mathbf{E}} \in \mathbb{R}^{(2n_c+n)\times(2n_c+n)}$, and $\bar{\mathbf{c}} \in \mathbb{R}^{2n_c+n}$ by

$$\mathbf{C} = \begin{pmatrix} \mathbf{T} \\ \mathbf{N} \\ \mathbf{B} \end{pmatrix}, \quad \bar{\mathbf{E}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} \end{pmatrix}, \quad \bar{\mathbf{c}} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{c} \end{pmatrix}.$$

Further we define the quadratic function $q : \mathbb{R}^{2n_c+n} \to \mathbb{R}$ by

$$q(\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{\mu}^T\mathbf{F}\boldsymbol{\mu} - \mathbf{d}^T\boldsymbol{\mu},$$

where $\mathbf{F} = \mathbf{C}\mathbf{A}_\kappa^{-1}\mathbf{C}^T + \bar{\mathbf{E}}$ and $\mathbf{d} = \mathbf{C}\mathbf{A}^{-1}\mathbf{l} - \bar{\mathbf{c}}$. The gradient $\nabla q : \mathbb{R}^{2n_c+n} \to \mathbb{R}^{2n_c+n}$ at $\boldsymbol{\mu} \in \mathbb{R}^{2n_c+n}$ reads as

$$\nabla q(\boldsymbol{\mu}) = \mathbf{F}\boldsymbol{\mu} - \mathbf{d}.$$

The following result holds.

Theorem 4.1 *The first component of the solution to* (4.1) *is given by:*

$$\mathbf{u} = \mathbf{A}^{-1}(\mathbf{l} - \mathbf{C}^T\boldsymbol{\lambda}). \tag{4.2}$$

The second component of the solution to (4.1) *solves this problem:*

$$\textit{Find } \boldsymbol{\lambda} \in \boldsymbol{\Lambda} \textit{ such that } 0 \leq (\boldsymbol{\mu} - \boldsymbol{\lambda})^T\nabla q(\boldsymbol{\lambda}) \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}. \tag{4.3}$$

Proof. In new notation:

$$L(\mathbf{v},\boldsymbol{\mu}) = \frac{1}{2}\mathbf{v}^T\mathbf{A}_\kappa\mathbf{v} - \mathbf{l}^T\mathbf{v} - \frac{1}{2}\boldsymbol{\mu}^T\bar{\mathbf{E}}\boldsymbol{\mu} - \boldsymbol{\mu}^T\bar{\mathbf{c}} + \boldsymbol{\mu}^T\mathbf{C}\mathbf{v}.$$

Since the second inequality in (4.1) is the unconstrained minimization, we get:

$$\frac{\partial L}{\partial \mathbf{v}}(\mathbf{u},\boldsymbol{\lambda}) = \mathbf{0} \;\Rightarrow\; \mathbf{A}_\kappa\mathbf{u} - \mathbf{l} + \mathbf{C}^T\boldsymbol{\lambda} = \mathbf{0}.$$

This proves (4.2), since $\mathbf{A}_\kappa$ is symmetric, positive definite. The second inequality in (4.1) can be written as:

$$\left.\begin{array}{l}\text{Find } \boldsymbol{\lambda} \in \boldsymbol{\Lambda} \text{ such that} \\ 0 \le \frac{1}{2}\boldsymbol{\mu}^T\bar{\mathbf{E}}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\lambda}^T\bar{\mathbf{E}}\boldsymbol{\lambda} + (\boldsymbol{\mu}-\boldsymbol{\lambda})^T\bar{\mathbf{c}} - (\boldsymbol{\mu}-\boldsymbol{\lambda})^T\mathbf{C}\mathbf{u} \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}.\end{array}\right\}$$

If we use the identity

$$\frac{1}{2}\boldsymbol{\mu}^T\bar{\mathbf{E}}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\lambda}^T\bar{\mathbf{E}}\boldsymbol{\lambda} = \frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\lambda})^T\bar{\mathbf{E}}(\boldsymbol{\mu}-\boldsymbol{\lambda}) + (\boldsymbol{\mu}-\boldsymbol{\lambda})^T\bar{\mathbf{E}}\boldsymbol{\lambda}$$

and substitute from (4.2), we get:

$$\left.\begin{array}{l}\text{Find } \boldsymbol{\lambda} \in \boldsymbol{\Lambda} \text{ such that} \\ 0 \le \frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\lambda})^T\bar{\mathbf{E}}(\boldsymbol{\mu}-\boldsymbol{\lambda}) + (\boldsymbol{\mu}-\boldsymbol{\lambda})^T(\bar{\mathbf{c}} - \mathbf{C}\mathbf{A}^{-1}\mathbf{l} + \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T\boldsymbol{\lambda} + \bar{\mathbf{E}}\boldsymbol{\lambda}) \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}.\end{array}\right\}$$

It is equivalent to the problem:

$$\left.\begin{array}{l}\text{Find } \boldsymbol{\lambda} \in \boldsymbol{\Lambda} \text{ such that} \\ 0 \le \frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\lambda})^T\bar{\mathbf{E}}(\boldsymbol{\mu}-\boldsymbol{\lambda}) + (\boldsymbol{\mu}-\boldsymbol{\lambda})^T\nabla q(\boldsymbol{\lambda}) \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}.\end{array}\right\} \tag{4.4}$$

Now we show that (4.3) and (4.4) have the same solution. If $\boldsymbol{\lambda}$ solves (4.3), then it solves (4.4) as $\bar{\mathbf{E}}$ is symmetric, positive semidefinite. The opposite implication will be proven by contradiction. Let $\bar{\boldsymbol{\lambda}} \in \boldsymbol{\Lambda}$ solve (4.4), but not (4.3). Then exists $\bar{\boldsymbol{\mu}} \in \boldsymbol{\Lambda}$ such that $0 > (\bar{\boldsymbol{\mu}} - \bar{\boldsymbol{\lambda}})^T\nabla q(\bar{\boldsymbol{\lambda}})$. Define $\boldsymbol{\mu}_k = k\bar{\boldsymbol{\mu}} + (1-k)\bar{\boldsymbol{\lambda}}$. Because $\boldsymbol{\Lambda}$ is convex, it holds $\boldsymbol{\mu}_k \in \boldsymbol{\Lambda}$ for $k \in [0,1]$ and $\boldsymbol{\mu}_k - \bar{\boldsymbol{\lambda}} = k(\bar{\boldsymbol{\mu}} - \bar{\boldsymbol{\lambda}})$. Using $\bar{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\lambda}}$ in (4.4), we get

$$0 \le k^2 e + kf \quad \forall k \in [0,1], \tag{4.5}$$

where $e = \frac{1}{2}(\bar{\boldsymbol{\mu}}-\bar{\boldsymbol{\lambda}})^T\bar{\mathbf{E}}(\bar{\boldsymbol{\mu}}-\bar{\boldsymbol{\lambda}}) \ge 0$ and $f = (\bar{\boldsymbol{\mu}}-\bar{\boldsymbol{\lambda}})^T\nabla q(\bar{\boldsymbol{\lambda}}) < 0$. If $e = 0$, we get the contradiction with (4.5). If $e > 0$, we get the contradiction with (4.5) for $0 < k < \min\{1, -f/e\}$. ∎

The problem (4.3) is the variational inequality characterizing the solution to the minimization

problem [5]:

$$\text{Find } \boldsymbol{\lambda} \in \boldsymbol{\Lambda} \text{ such that } q(\boldsymbol{\lambda}) \leq q(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \boldsymbol{\Lambda}. \tag{4.6}$$

This problem may be solved by an appropriate optimalization algorithm [9]. After obtaining $\boldsymbol{\lambda}$, we can get $\mathbf{u}$ from (4.2).

Let us note, that (4.6) is the minimization of the strictly convex quadratic function on the convex set that guarantees the existence of the unique solution $\boldsymbol{\lambda}$.Then the component $\mathbf{u}$ is uniquely determined by (4.2). This analysis proves the existence and the uniqueness of the solution to (4.1) and of all previous algebraic problems.

4.2 Optimality conditions

The solution to (4.1) can be uniquely characterized by a system of equalities and inequalities called the *optimality conditions* that are summarized in the following theorem.

Theorem 4.2 *The pair* $(\mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{2(n-n_d)} \times \mathbb{R}^{2n_c+n}$*, where* $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_t^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T)^T$*, is the solution to* (4.1) *iff:*

$$\mathbf{A}\mathbf{u} - \mathbf{l} + \mathbf{T}^T\mathbf{s}_t + \mathbf{N}^T\boldsymbol{\lambda}_n + \mathbf{B}^T\mathbf{p} = \mathbf{0}, \tag{4.7}$$

$$\mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c} = \mathbf{0}, \tag{4.8}$$

$$\mathbf{N}\mathbf{u} = \mathbf{0} \tag{4.9}$$

$$\left.\begin{array}{l} (\mathbf{T}\mathbf{u})_i = 0 \Rightarrow |s_{ti}| \leq g_i, \\ (\mathbf{T}\mathbf{u})_i > 0 \Rightarrow s_{ti} = g_i + \kappa_i(\mathbf{T}\mathbf{u})_i, \\ (\mathbf{T}\mathbf{u})_i < 0 \Rightarrow s_{ti} = -g_i + \kappa_i(\mathbf{T}\mathbf{u})_i, \end{array}\right\} \; i \in \mathcal{N}, \tag{4.10}$$

where $\mathbf{s}_t = \boldsymbol{\lambda}_t + \mathbf{D}(\boldsymbol{\kappa})\mathbf{T}\mathbf{u}$ *and* $\mathcal{N} = \{1, ..., n_c\}$.

Proof. The second inequality in (4.1) is equivalent to

$$\mathbf{A}_\kappa\mathbf{u} - \mathbf{l} + \mathbf{T}^T\boldsymbol{\lambda}_t + \mathbf{N}^T\boldsymbol{\lambda}_n + \mathbf{B}^T\mathbf{p} = \mathbf{0}.$$

Using $\mathbf{A}_\kappa = \mathbf{A} + \mathbf{T}^T\mathbf{D}(\boldsymbol{\kappa})\mathbf{T}$, we get

$$\mathbf{A}\mathbf{u} - \mathbf{l} + \mathbf{T}^T(\boldsymbol{\lambda}_t + \mathbf{D}(\boldsymbol{\kappa})\mathbf{T}\mathbf{u}) + \mathbf{N}^T\boldsymbol{\lambda}_n + \mathbf{B}^T\mathbf{p} = \mathbf{0}$$

that is (4.7). The first inequality in (4.1) holds, if the seperate inequalities for $\boldsymbol{\mu} = (\mathbf{0}^T, \mathbf{0}^T, \mathbf{q}^T)^T$, $\boldsymbol{\mu} =$

$(\mathbf{0}^T, \boldsymbol{\mu}_n^T, \mathbf{0}^T)^T$, and $\boldsymbol{\mu} = (\boldsymbol{\mu}_t^T, \mathbf{0}^T, \mathbf{0}^T)^T$ are satisfied separately:

$$-\frac{1}{2}\mathbf{q}^T\mathbf{E}\mathbf{q} - \mathbf{c}^T\mathbf{q} + \mathbf{q}^T\mathbf{B}\mathbf{u} \le -\frac{1}{2}\mathbf{p}^T\mathbf{E}\mathbf{p} - \mathbf{c}^T\mathbf{p} + \mathbf{p}^T\mathbf{B}\mathbf{u} \quad \forall \mathbf{q} \in \mathbb{R}^n, \tag{4.11}$$

$$\boldsymbol{\mu}_n^T\mathbf{N}\mathbf{u} \le \boldsymbol{\lambda}_n^T\mathbf{N}\mathbf{u} \quad \forall \boldsymbol{\mu}_n \in \mathbb{R}^{n_c}, \tag{4.12}$$

$$\boldsymbol{\mu}_t^T\mathbf{T}\mathbf{u} \le \boldsymbol{\lambda}_t^T\mathbf{T}\mathbf{u} \quad \forall \boldsymbol{\mu}_t \in \boldsymbol{\Lambda}_t, \tag{4.13}$$

with $\boldsymbol{\lambda}_t \in \boldsymbol{\Lambda}_t$. We will analyze (4.11) using the same technique as in the proof of Theorem 4.1. From (4.11) we get:

$$0 \le \frac{1}{2}(\mathbf{p}-\mathbf{q})^T\mathbf{E}(\mathbf{p}-\mathbf{q}) + (\mathbf{p}-\mathbf{q})^T(\mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c}) \quad \forall \mathbf{q} \in \mathbb{R}^n.$$

The last inequality holds iff

$$0 \le (\mathbf{p}-\mathbf{q})^T(\mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c}) \quad \forall \mathbf{q} \in \mathbb{R}^n$$

that is equivalent to (4.8). (4.12) is satisfied iff (4.9) holds. The inequality (4.13) holds iff the following seperate inequalities hold:

$$0 \le (\lambda_{ti} - \mu_{ti})(\mathbf{T}\mathbf{u})_i \quad \forall \mu_{ti} \in \mathbb{R},\ |\mu_{ti}| \le g_i$$

with $|\lambda_{ti}| \le g_i$ for $i \in \mathcal{N}$. We get:

$$\left.\begin{aligned} &(\mathbf{T}\mathbf{u})_i = 0 \Rightarrow |\lambda_{ti}| \le g_i, \\ &(\mathbf{T}\mathbf{u})_i > 0 \Rightarrow \lambda_{ti} = g_i, \\ &(\mathbf{T}\mathbf{u})_i < 0 \Rightarrow \lambda_{ti} = -g_i, \end{aligned}\right\}\ i \in \mathcal{N}.$$

Using $s_{ti} = \lambda_{ti} + \kappa_i(\mathbf{T}\mathbf{u})_i$ we arrive at (4.10). ∎

The relation between s_{ti} and $(\mathbf{T}\mathbf{u})_i$ in (4.10) represents the continuous piecewise linear function (see Figure 3):

$$(\mathbf{T}\mathbf{u})_i = \begin{cases} \kappa_i^{-1}(s_{ti} + g_i), & s_{ti} \le -g_i, \\ 0, & s_{ti} \in (-g_i, g_i), \\ \kappa_i^{-1}(s_{ti} - g_i), & s_{ti} \ge g_i. \end{cases} \tag{4.14}$$

This function can be written using the max-function:

$$\phi : \mathbb{R} \to \mathbb{R},\ \phi(y) = \max\{0, y\},\ y \in \mathbb{R}$$

so that

$$(\mathbf{T}\mathbf{u})_i = \phi(\kappa_i^{-1}(s_{ti} - g_i)) - \phi(-\kappa_i^{-1}(s_{ti} + g_i)). \tag{4.15}$$

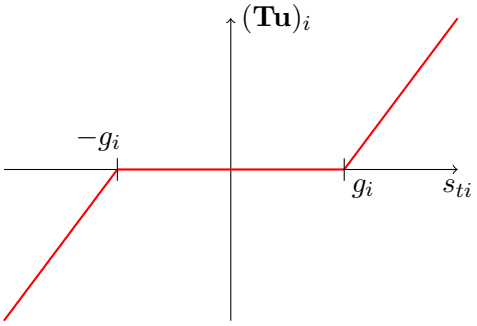

Figure 3: $(\mathbf{Tu})_i$ versus s_{ti}

The following lemma sumarizes (4.10) in one equation.

Lemma 4.3 *The relations* (4.10) *are satisfied iff*

$$\mathbf{\Phi}(\mathbf{u}, \mathbf{s}_t) = \mathbf{0},$$

where $\mathbf{\Phi}(\mathbf{u}, \mathbf{s}_t) = \mathbf{Tu} - \boldsymbol{\phi}(\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t - \mathbf{g})) + \boldsymbol{\phi}(-\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t + \mathbf{g}))$ *and* $\boldsymbol{\phi} : \mathbb{R}^{n_c} \to \mathbb{R}^{n_c}$ *is defined by* $\boldsymbol{\phi}(\mathbf{y}) = (\phi(y_1), ..., \phi(y_{n_c}))^T,\ \mathbf{y} \in \mathbb{R}^{n_c}$.

Proof. It follows from (4.15) ∎

Let $N = 2(n - n_d) + n + 2n_c$ and let the function

$$\mathbf{G} : \mathbb{R}^N \to \mathbb{R}^N, \quad \mathbf{y} = (\mathbf{u}^T, \mathbf{s}_t^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T)^T \in \mathbb{R}^N$$

be defined by

$$\mathbf{G}(\mathbf{y}) = \begin{pmatrix} \mathbf{Au} - \mathbf{l} + \mathbf{T}^T\mathbf{s}_t + \mathbf{N}^T\boldsymbol{\lambda}_n + \mathbf{B}^T\mathbf{p} \\ \mathbf{\Phi}(\mathbf{u}, \mathbf{s}_t) \\ \mathbf{Nu} \\ \mathbf{Bu} - \mathbf{Ep} - \mathbf{c} \end{pmatrix}. \tag{4.16}$$

Theorem 4.4 *The pair* $(\mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{2(n-n_d)} \times \mathbf{\Lambda}$ *solves* (4.1) *iff* $\mathbf{y} = (\mathbf{u}^T, \mathbf{s}_t^T, \boldsymbol{\lambda}_n^T, \mathbf{p}^T)^T \in \mathbb{R}^N$ *solves*

$$\mathbf{G}(\mathbf{y}) = \mathbf{0}, \tag{4.17}$$

where $\boldsymbol{\lambda}_t = \mathbf{s}_t - \mathbf{D}(\boldsymbol{\kappa})\mathbf{Tu}$.

Proof. (4.16) is equivalent to the optimality conditions (4.7) - (4.10). ∎

The equation (4.17) can be solved by a Newton-type method. Due to the presence of the max-functions, the Jacobi matrix to $\mathbf{G}$ does not exists at all points so that the classical Newton

method can not be used. Fortunately, $\mathbf{G}$ is semi-smooth in the sense of the next chapter so that the semi-smooth Newton method can be used.

5 Semi-smooth Newton method

The SSNM uses a slant differentiability of functions. After recalling this concept [6, 8], we will apply it to the equation (4.17).

5.1 An abstract setting

Let Y, Z be Banach spaces with norms $||\cdot||_Y$, $||\cdot||_Z$, respectively. Let $U \subseteq Y$ be an open subset and $G : U \to Z$ be a mapping.

Definition 5.1 *(a) The mapping G is called slantly differentiable at $y \in U$, if there exists a system of mappings $G^o : U \to \mathcal{L}(Y, Z)$ such that $\{G^o(y+h)\}$ is uniformly bounded for sufficiently small $h \in Y$ and*

$$\lim_{h\to 0} \frac{||G(y+h) - G(y) - G^o(y+h)h||_Z}{||h||_Y} = 0.$$

Then G^o is called a slanting mapping for G at y.
(b) G is called slantly differentiable in U, if there exists $G^o : U \to \mathcal{L}(Y, Z)$ such that G^o is a slanting mapping for G at every point $y \in U$. Then G^o is termed a slanting mapping for G in U.

Definition 5.2 *Assume that*

$$\lim_{k\to\infty} y^k = y^*, \quad y^k, y^* \in Y.$$

Then the convergence is superlinear, if

$$\lim_{k\to\infty} \frac{||y^{k+1} - y^*||_Y}{||y^k - y^*||_Y} = 0.$$

Theorem 5.1 *Let G be slantly differentiable in U with a slanting mapping G^o. Suppose that $y^* \in U$ is a solution to the nonlinear equation $G(y) = 0$. If $G^o(y)$ is injective for all $y \in U$ and $\{||G^o(y)^{-1}||_{\mathcal{L}(Z,Y)} : y \in U\}$ is bounded, then the Newton iterations*

$$y^{k+1} = y^k - G^o(y^k)^{-1} G(y^k) \tag{5.1}$$

converge superlinearly to y^* provided that $||y^0 - y^*||_Y$ is sufficiently small.

Proof. See [6, 8]. ∎

The slanting mapping will be termed the slanting function, since we deal with finite dimensional spaces. The following example plays an important role in our analysis.

Example 1
The max-function $\phi(y) = \max\{0, y\}$ is slantly differentiable in $\mathbb{R}$ and

$$\phi^o(y) = \begin{cases} 1 & \text{for } y > 0, \\ \eta & \text{for } y = 0, \\ 0 & \text{for } y < 0, \end{cases}$$

is the slanting function for an arbitrary $\eta \in \mathbb{R}$ (we use $\eta = 1$ in our computations). ■

5.2 Active/inactive set implementation

According to (4.16) the function $\mathbf{G}$ in (4.17) has the block structure:

$$\mathbf{G}(\mathbf{y}) = (\mathbf{G}_1(\mathbf{y})^T, \mathbf{G}_2(\mathbf{y})^T, \mathbf{G}_3(\mathbf{y})^T, \mathbf{G}_4(\mathbf{y})^T)^T.$$

The first component $\mathbf{G}_1(\mathbf{y}) = \mathbf{A}\mathbf{u} - \mathbf{l} + \mathbf{T}^T\mathbf{s}_t + \mathbf{N}^T\boldsymbol{\lambda}_n + \mathbf{B}^T\mathbf{p}$ is differentiable. Thus its classical derivative is the slanting function:

$$\mathbf{G}_1^o(\mathbf{y}) = (\mathbf{A}, \mathbf{T}^T, \mathbf{N}^T, \mathbf{B}^T).$$

Since $\mathbf{G}_2(\mathbf{y}) = \mathbf{T}\mathbf{u} - \boldsymbol{\phi}(\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t - \mathbf{g})) + \boldsymbol{\phi}(-\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t + \mathbf{g}))$ is defined by the max-functions, the slanting function $\mathbf{G}_2^o$ follows from Example 1. Its convenient setting uses active/inactive sets. Let $\mathcal{A}_t = \mathcal{A}_t(\mathbf{y}), \mathcal{I}_t^- = \mathcal{I}_t^-(\mathbf{y}),$ and $\mathcal{I}_t^+ = \mathcal{I}_t^+(\mathbf{y})$ be the active and inactive sets at $\mathbf{y} \in \mathbb{R}^N$ defined by:

$$\begin{aligned} \mathcal{A}_t &= \{i \in \mathcal{N} : s_{ti} \in (-g_i, g_i)\}, \\ \mathcal{I}_t^+ &= \{i \in \mathcal{N} : s_{ti} \geq g_i\}, \\ \mathcal{I}_t^- &= \{i \in \mathcal{N} : s_{ti} \leq -g_i\}, \end{aligned}$$

where $\mathcal{N} = \{1, ..., n_c\}$. Let us define the indicator matrix to $\mathscr{S} \subseteq \mathcal{N}$ by $D(\mathscr{S}) = \text{diag}(s_1, ..., s_{n_c}) \in \mathbb{R}^{n_c \times n_c}$ by $s_i = 1$ for $i \in \mathscr{S}$ and $s_i = 0$ if $i \notin \mathscr{S}$. The slanting function with respect to the variable $\mathbf{u}$ is the classical derivative, i.e.:

$$\frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{u}} = \mathbf{T}.$$

Using indicator matrices, one can write:

$$\mathbf{G}_2(\mathbf{y}) = \mathbf{T}\mathbf{u} - \mathbf{D}(\mathcal{I}_t^+)(\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t - \mathbf{g})) + \mathbf{D}(\mathcal{I}_t^-)(-\mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{s}_t + \mathbf{g})).$$

Differentiating with respect to $\mathbf{s}_t$ by the standard differentiation rules, we get:

$$\frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{s}_t} = -\mathbf{D}(\mathcal{I}_t^+)\mathbf{D}(\boldsymbol{\kappa})^{-1} - \mathbf{D}(\mathcal{I}_t^-)\mathbf{D}(\boldsymbol{\kappa})^{-1} = -\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-).$$

We obtain

$$\mathbf{G}_2^o(\mathbf{y}) = (\mathbf{T}, -\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-), \mathbf{0}, \mathbf{0}).$$

Remaining two components $\mathbf{G}_3(\mathbf{y}) = \mathbf{N}\mathbf{u}$ and $\mathbf{G}_4(\mathbf{y}) = \mathbf{B}\mathbf{u} - \mathbf{E}\mathbf{p} - \mathbf{c}$ are differentiable in the classical sense so that:

$$\mathbf{G}_3^o(\mathbf{y}) = (\mathbf{N}, \mathbf{0}, \mathbf{0}, \mathbf{0}),$$

$$\mathbf{G}_4^o(\mathbf{y}) = (\mathbf{B}, \mathbf{0}, \mathbf{0}, -\mathbf{E}).$$

Summarizing the previous results, we arrive at the following slanting function for $\mathbf{G}$:

$$\mathbf{G}^o(\mathbf{y}) = \left(\begin{array}{c|ccc} \mathbf{A} & \mathbf{T}^T & \mathbf{N}^T & \mathbf{B}^T \\ \hline \mathbf{T} & -\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-) & \mathbf{0} & \mathbf{0} \\ \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & -\mathbf{E} \end{array}\right)$$

Each iterative step of (5.1) leads to the linear system for $\mathbf{y}^{k+1}$ with the matrix $\mathbf{G}^o(\mathbf{y}^k)$ and the right-hand side vector $\mathbf{G}^o(\mathbf{y}^k)\mathbf{y}^k - \mathbf{G}(\mathbf{y}^k)$. This vector may be simplified as follows:

$$\mathbf{G}^o(\mathbf{y}^k)\mathbf{y}^k - \mathbf{G}(\mathbf{y}^k) = \begin{pmatrix} \mathbf{l} \\ \mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{D}(\mathcal{I}_t^-) - \mathbf{D}(\mathcal{I}_t^+))\mathbf{g} \\ \mathbf{0} \\ \mathbf{c} \end{pmatrix}.$$

We arrive at the following active/inactive set implementation of (5.1) that uses only inactive sets.

Algorithm SSNM

Given $\mathbf{y}^0 = ((\mathbf{u}^0)^T, (\mathbf{s}_t^0)^T, (\boldsymbol{\lambda}_n^0)^T, (\mathbf{p}^0)^T)^T \in \mathbb{R}^N$. For $k \geq 0$ compute:

(Step 1) Assembly the inactive sets at $\mathbf{y}^k = ((\mathbf{u}^k)^T, (\mathbf{s}_t^k)^T, (\boldsymbol{\lambda}_n^k)^T, (\mathbf{p}^k)^T)^T$:

$$\mathcal{I}_t^+ = \{i \in \mathcal{N} : s_{ti}^k \geq g_i\} \quad \text{and} \quad \mathcal{I}_t^- = \{i \in \mathcal{N} : s_{ti}^k \leq -g_i\}.$$

(Step 2) Solve the linear system

$$\left(\begin{array}{c|ccc} \mathbf{A} & \mathbf{T}^T & \mathbf{N}^T & \mathbf{B}^T \\ \hline \mathbf{T} & -\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-) & \mathbf{0} & \mathbf{0} \\ \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & -\mathbf{E} \end{array}\right) \begin{pmatrix} \mathbf{u}^{k+1} \\ \mathbf{s}_t^{k+1} \\ \boldsymbol{\lambda}_n^{k+1} \\ \mathbf{p}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{l} \\ \mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{D}(\mathcal{I}_t^-) - \mathbf{D}(\mathcal{I}_t^+))\mathbf{g} \\ \mathbf{0} \\ \mathbf{c} \end{pmatrix}.$$

Let us note that the results of Theorem 5.1, including the superlinear convergence rate, hold if the linear systems in *Step 2* are solved exactly. Unfortunately, exact solution of large linear systems is unrealistic.

5.3 Inexact implementation

The efficient implementation of ALGORITHM SSNM depends on the way how the inner linear systems are solved. For the sake of simplicity we introduce the matrices:

$$\mathbf{C} = \begin{pmatrix} \mathbf{T} \\ \mathbf{N} \\ \mathbf{B} \end{pmatrix}, \quad \bar{\mathbf{E}}^k = \begin{pmatrix} \mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} \end{pmatrix},$$

and the vectors:

$$\mathbf{r}^{k+1} = \begin{pmatrix} \mathbf{s}_t^{k+1} \\ \boldsymbol{\lambda}_n^{k+1} \\ \mathbf{p}^{k+1} \end{pmatrix}, \quad \mathbf{h}^k = \begin{pmatrix} \mathbf{D}(\boldsymbol{\kappa})^{-1}(\mathbf{D}(\mathcal{I}_t^-) - \mathbf{D}(\mathcal{I}_t^+))\mathbf{g} \\ \mathbf{0} \\ \mathbf{c} \end{pmatrix}.$$

The linear system from *Step 2* takes the form:

$$\begin{pmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & -\bar{\mathbf{E}}^k \end{pmatrix} \begin{pmatrix} \mathbf{u}^{k+1} \\ \mathbf{r}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{l} \\ \mathbf{h}^k \end{pmatrix}.$$

The Schur complement system reads as follows:

$$(\mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T + \bar{\mathbf{E}}^k)\mathbf{r}^{k+1} = \mathbf{C}\mathbf{A}^{-1}\mathbf{l} - \mathbf{h}^k. \tag{5.2}$$

After its solving, one can compute $\mathbf{u}^{k+1}$ using

$$\mathbf{u}^{k+1} = \mathbf{A}^{-1}(\mathbf{l} - \mathbf{C}^T\mathbf{r}^{k+1}). \tag{5.3}$$

We will use the conjugate gradient method as the inner solver for solving (5.2). If we look closely at each member of (5.2) and (5.3), we can see that some of them will remain the same in each iteration and we can calculate them beforehand. These members are matrix $\mathbf{C}$, vector $\mathbf{C}\mathbf{A}^{-1}\mathbf{l}$ and the inverse matrix to $\mathbf{A}$. To make the calculations easier and in cosideration of the properties of $\mathbf{A}$, we will not have to work with the inverse directly, but we will use the Cholesky decomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is lower triangular. The action of $\mathbf{A}^{-1}$ on the vector $\mathbf{v}$ is computed by two backward substitutions as $\mathbf{L}^{-T}(\mathbf{L}^{-1}\mathbf{v})$. Another thing we can notice is that the vector $\mathbf{u}^{k+1}$ changes only with the changes of the vector $\mathbf{r}^{k+1}$, while having no effect on the remaining steps. It is then enough to calculate $\mathbf{u}^{k+1}$ only once at the end of the iterative process for the last value of k.

Once we have all the matrices and vectors ready, we can start with the SSNM iterations. As the first step, we create the inactive sets $\mathcal{I}_t^+$ and $\mathcal{I}_t^-$. The matrix $\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-)$ may be represented by its diagonal $\mathbf{d}^k = \mathrm{diag}(\mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-))$. Note that $\mathbf{d}^k$ is the only part of $\bar{\mathbf{E}}^k$ that depends on the iteration. As it was mentioned earlier, we use the conjugate gradient method to calculate $\mathbf{r}^{k+1}$. It is referred by

$$\mathbf{r}^{k+1} = \ \mathrm{CGM}(\mathbf{L}, \mathbf{C}, \bar{\mathbf{E}}^k, \mathbf{b}^k, \mathbf{r}^k, cgmtol^k, numit),$$

where $\mathbf{S}^k = \mathbf{C}\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{C}^T + \bar{\mathbf{E}}^k$ is the Shur complement from (5.2), $\mathbf{b}^k = \mathbf{C}\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{l} - \mathbf{h}^k$, $\mathbf{r}^k$ is the initial CGM iteration taken as the result from the previous Newton step, $cgmtol^k$ is the stopping tolerance, and $numit$ is the stopping number of iterations.

The Algorithm ISSNM summarizes the implementation details discussed above. Moreover we use the adaptive stopping tolerance in *Step 2.2* for inexact solving of inner linear systems (5.2). The value $cgmtol^{k+1}$ respects the precission err^k achieved on the global level and, if the progress is not sufficient, it uses improved inner tolerance from the previous Newton step.

Algorithm ISSNM

%*Preparation for SSNM and CGM iterations:*
(Step 1) Given $\mathbf{r}^0 \in \mathbb{R}^{n+2n_c}$, $tol > 0$, and r_{tol},$c_{fact} \in (0,1)$.
(Step 1.1) Set $err^0 = 1$, $k = 0$, and $cgmtol^0 = r_{tol}/c_{fact}$.
(Step 1.2) Compute $\mathbf{L} = \mathrm{chol}(\mathbf{A})$, create $\mathbf{C}$, and calculate $\mathbf{b} = \mathbf{C}\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{l}$.

% *SSNM iterations and calculation of* $\mathbf{u}$*:*
(Step 2) While $err^k > tol$, go to *Step 2.1*, else set $\mathbf{r} = \mathbf{r}^k$, calculate $\mathbf{u} = \mathbf{u}^k$ from (5.3) with $k+1$ replaced by k, return $\mathbf{y} = (\mathbf{u}^T, \mathbf{r}^T)^T$, and stop.
(Step 2.1) Assemble the inactive sets $\mathcal{I}_t^+$ and $\mathcal{I}_t^-$ at $\mathbf{r}^k$, create $\mathbf{d}^k$ defining $\bar{\mathbf{E}}^k$, and $\mathbf{b}^k = \mathbf{b} - \mathbf{h}^k$.
(Step 2.2) $cgmtol^{k+1} = \min(r_{tol} \times err^k, c_{fact} \times tol^k)$.
(Step 2.3) $\mathbf{r}^{k+1} = \ \mathrm{CGM}(\mathbf{L}, \mathbf{C}, \bar{\mathbf{E}}^k, \mathbf{b}^k, \mathbf{r}^k, cgmtol^{k+1}, numit)$.
(Step 2.4) $err^{k+1} = ||\mathbf{r}^{k+1} - \mathbf{r}^k||/(||\mathbf{r}^{k+1}|| + 1)$, $k = k+1$ and go to *Step 2.*

5.4 Preconditioning

Let us note that the Shur complement matrix of the linear system (5.2), i.e.,

$$\mathbf{S} = \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T + \bar{\mathbf{E}}^k,$$

can be ill-conditioned especially, when κ is small, since some diagonal entires of $\bar{\mathbf{E}}^k$ are very large. We will use the diagonal preconditioner:

$$\mathbf{P} = \mathrm{diag}(\mathbf{S}).$$

The following theorem shows that the spectral condition number of the preconditioned matrix $\mathbf{P}^{-1}\mathbf{S}$ is independent on κ.

Theorem 5.2 *Let* $\mathbf{D}^k = \mathbf{D}(\boldsymbol{\kappa})^{-1}\mathbf{D}(\mathcal{I}_t^+ \cup \mathcal{I}_t^-)$. *It holds:*

$$cond(\mathbf{P}^{-1}\mathbf{S}) \leq cond(\mathbf{S} - \mathbf{D}^k)cond(\mathbf{P} - \mathbf{D}^k).$$

Proof. It is a simple modification of the analogous result from [9]. ∎

Since the first term in $\mathbf{S}$ is not given explicitely, we will use the approximation of the preconditioner:

$$\mathbf{P} \approx \mathrm{diag}(\mathbf{C}\mathrm{diag}(\mathbf{A})^{-1}\mathbf{C}^T) + \mathrm{diag}(\bar{\mathbf{E}}^k).$$

5.5 MATLAB implementation

Our solver is implemented in MATLAB. It is divided into three sections: preprocessing section, solver section, and postprocessing section. The preprocessing section is not described here, it deals with generating matrices and vectors, which are saved in a global data structure. The solver section containing the MATLAB implementation of the ALGORITHM ISSNM will be described with all details. Finally, from the postprocessing section we will use graphical representations of computed solutions, computed complexity characteristics, etc.

5.5.1 Global data structures

In the preprocessing section we generate the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{T}$, $\mathbf{N}$, and $\mathbf{E}$ together with the vectors $\mathbf{l}$, $\mathbf{c}$, $\mathbf{g}$, and $\boldsymbol{\kappa}$. All are stored in global structure called data:

$\mathbf{A}$ = data.A
$\mathbf{B}$ = data.B
$\mathbf{E}$ = data.E
$\mathbf{T}$ = data.T
$\mathbf{N}$ = data.N
$\mathbf{l}$ = data.l
$\mathbf{c}$ = data.c
$\mathbf{g}$ = data.g
$\boldsymbol{\kappa}$ =data.kappa

This allows us to access the data directly when needed. Other objects will be added into the structure data in the solver section.

The global structure opts is then used for parameters driving the ALGORITHM ISSNM. This data structure contains the following parameters:

$tol =$ opts. tol

$r_{tol} =$ opts. rtol

$c_{fact} =$ opts. cfact

$numit =$ opts.max_it_inner

opts.max_it is the stopping number of the SSNM iterations

5.5.2 Solver section: SSNM implementation

We will discuss the implementation of ALGORITHM ISSNM. Having all the data prepared in the data structure, we can immediately start. We begin with the initial SSNM iteration $\mathbf{r}^0 = \mathbf{0}$. We use the MATLAB function chol to get the Cholesky factor $\mathbf{L}$ that is added to the structure data.

The SSNM iterations in ALGORITHM ISSNM are realized by the while loop. After assembling the inactive sets $\mathcal{I}_t^+$ and $\mathcal{I}_t^-$, we create diagonal entries of $\mathbf{D}(\boldsymbol{\kappa})\mathbf{D}(\mathcal{I}_t^+)$ and $\mathbf{D}(\boldsymbol{\kappa})\mathbf{D}(\mathcal{I}_t^-)$ in D_plus and D_minus, respectively. The vector $\mathbf{d}^k$ representing the block of $\bar{\mathbf{E}}^k$ is inserted in the structure data as data.Dpm = D_plus + D_minus. Note that [data.T; data.N; data.B] is the matrix $\mathbf{C}$.

```
function SemiSmoothNewton
global data opts

% Preparing for SSNM, CGM, and preconditioner
nu=size(data.l,1); nc=size(data.g,1);
data.L=chol(data.A,'lower');
b0=[data.T;data.N;data.B]*((data.L\data.l)'/data.L)';
Prec=diag([data.T,data.N,data.B]*spdiags(1./diag(data.A),0,nu,nu)*...
[data.T,data.N,data.B]') + [zeros(2*nc,1);diag(data.E)];

% SSNM iterations
err=1; k=0; r0=zeros(size(data.c,1)+2*nc,1);
cgmtol=opts.rtol/opts.cfact;
while k<opts.max_it && err>opts.tol

    I_plus =r0(1:nc,:)>=data.g; D_plus=zeros(nc,1);
      D_plus(I_plus)=1./data.kappa(I_plus);
    I_minus=r0(1:nc,:)<=-data.g; D_minus=zeros(nc,1);
      D_minus(I_minus)=1./data.kappa(I_minus);
    data.Dpm=D_plus+D_minus;
    data.P=Prec; data.P(1:nc)=data.P(1:nc)+data.Dpm;
```

```
    cgmtol=min([opts.rtol*err,opts.cfact*cgmtol]);
    b=b0-[(D_minus-D_plus).*data.g;zeros(nc,1);data.c];
    r=cgm(@S,b,r0,cgmtol,opts.max_it_inner,@Pm);

    err=norm(r-r0)/(norm(r)+1); r0=r; k=k+1;
end

% Computations of velocity, pressure and Lagrange multipliers
data.u=((data.L\(data.l-(r'*[data.T;data.N;data.B])'))'/data.L)';
data.lt=r(1:nc)-data.kappa.*(data.T*data.uu);
data.ln=r(nc+1:2*nc);
data.p=r(2*nc+1:end);
```

Code 1: Implementation of ALGORITHM ISSNM

5.5.3 CGM implementation

We use the standard implementation of the CGM [7] with the preconditioner $\mathbf{P}$. We get the unpreconditioned version of the CGM for $\mathbf{P} = \mathbf{I}$.

```
function [x,r,it]=cgm(A,b,x0,tol,max_it,P)

x = x0; r = b-A(x); z = P(r);
p = z; ro_k = r'*z;
it = 0; tol = tol*sqrt(b'*b);

while norm(r)>tol && it<max_it
   w = A(p);
   alpha = ro_k/(p'*w);
   x = x+alpha*p;
   r = r-alpha*w;
   z = P(r);
   ro_k_1 = ro_k;
   ro_k = r'*z;
   beta = ro_k/ro_k_1;
   p = z+beta*p;
   it = it+1;
end
```

Code 2: Conjugate gradient method implementation

The input parameter A is the function handle that is suplied in SemiSmoothNewton.m by the function S.m that efficiently computes actions of the Shur complements from (5.2) on vectors. The last two lines in the function S.m are calculations with $\bar{\mathbf{E}}^k$, where we use only its nonzero parts.

```
function y=S(x)
global data

nc=size(data.g,1);
y=(x'*[data.T;data.N;data.B])';
y=((data.L\y)'/data.L)';
y=[data.T;data.N;data.B]*y;
y(1:nc,1)=y(1:nc,1)+data.Dpm.*x(1:nc);
y(2*nc+1:end)=y(2*nc+1:end)+data.E*x(2*nc+1:end);
```

Code 3: Action of the Shur complement

The input parameter P is the function handle that represents the preconditioner. Its implementation based on Section 5.4 is computed by the following code in the function P.m.

```
function y=P(x)
global data

y=x./data.P;
```

Code 4: Action of the preconditioner

6 Numerical experiments

We will assess the performance of our algorithm for two different domains. First we will experimentaly find optimal values of r_{tol} and c_{fact} for the adaptive CGM tolerance. Then we will test what impact does κ have on calculations as it tends to 0 and the efficiency of the preconditioner.

In tables below we report $iter/n_A$, where $iter$ is the number of outer iterations, i.e. the last value of k, and n_A is the number of matrix-vector multiplications by $\mathbf{A}^{-1}$. Let us note that n_A determines complexity of computations. The values n_u, n_p, and n_c represent different discretizations, where n_u is the number of velocity components, n_p is the number of pressure components, and n_c is the number of nodes lying on $\bar{\gamma}_C \backslash \bar{\gamma}_D$.

6.1 Example 1

The first experiments will be done on the square domain defined as follows: $\Omega = (0,1) \times (0,1)$, $\gamma_D = (0,1) \times 1$, $\gamma_{N_{\text{left}}} = 0 \times (0,1)$, $\gamma_{N_{\text{right}}} = 1 \times (0,1)$, $\gamma_N = \gamma_{N_{\text{left}}} \cup \gamma_{N_{\text{right}}}$, $\gamma_C = (0,1) \times 0$, $\mathbf{u}_D = \mathbf{0}$, $\boldsymbol{\sigma}_N = \boldsymbol{\sigma}_{\text{exp}}$, $g = 10$, and $\nu = 1$, where $\mathbf{u}_{\text{exp}}(x,y) = (-\cos(2\pi x)\sin(2\pi y) + \sin(2\pi y), \sin(2\pi x)\cos(2\pi y) - \sin(2\pi x))$ and $\mathbf{p}_{\text{exp}}(x,y) = 2\pi(\cos(2\pi y) - \cos(2\pi x))$.

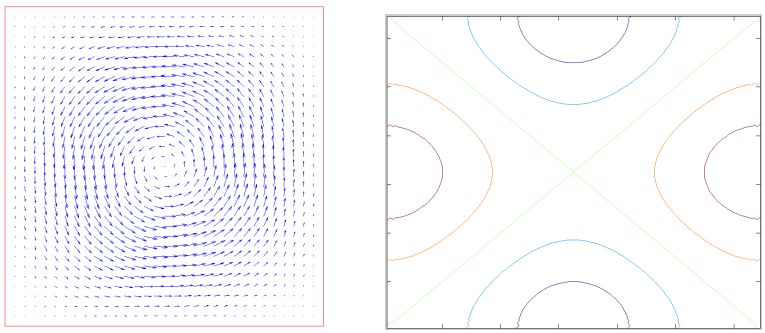

Figure 4: Velocity field and isobars

Tables 1 - 5 are computed with different values r_{tol}, while each column represents results for different value c_{fact}. Further, $\kappa = 100$ and $tol = 10^{-8}$. Although the computational complexities are similar, we decide to chose $r_{tol} = 0.9$ and $c_{fact} = 0.08$ as the optimal values.

$n_u/n_p/n_c$	$c_{fact} = 0.02$	$c_{fact} = 0.06$	$c_{fact} = 0.08$	$c_{fact} = 0.09$	$c_{fact} = 0.099$
544/289/17	8/582	10/549	10/498	11/555	11/525
2112/1089/33	8/1008	10/966	11/940	11/915	11/859
8320/4225/65	8/1461	10/1486	11/1422	12/1530	12/1519
33024/16641/129	8/2279	11/2557	12/2430	12/2371	12/2183
74112/37249/193	8/2939	11/3016	12/3128	12/2826	13/3214

Table 1: $r_{tol} = 0.01$

$n_u/n_p/n_c$	$c_{fact} = 0.02$	$c_{fact} = 0.06$	$c_{fact} = 0.08$	$c_{fact} = 0.09$	$c_{fact} = 0.099$
544/289/17	8/505	10/551	11/540	11/536	12/555
2112/1089/33	8/929	11/958	12/1042	12/950	12/886
8320/4225/65	9/1659	11/1490	12/1540	12/1425	13/1573
33024/16641/129	9/2618	11/2238	12/2232	13/2384	13/2240
74112/37249/193	9/3159	11/2921	13/3366	13/2943	13/2799

Table 2: $r_{tol} = 0.05$

$n_u/n_p/n_c$	$c_{fact} = 0.02$	$c_{fact} = 0.06$	$c_{fact} = 0.08$	$c_{fact} = 0.09$	$c_{fact} = 0.099$
544/289/17	8/504	10/488	11/525	12/564	12/528
2112/1089/33	9/1018	11/1002	12/937	12/922	13/960
8320/4225/65	9/1616	11/1520	12/1466	13/1629	13/1452
33024/16641/129	$9/x2578$	$11/x2181$	13/2514	13/2459	13/2258
74112/37249/193	$9/x3093$	$12/x3324$	13/3164	13/2799	14/3064

Table 3: $r_{tol} = 0.1$

$n_u/n_p/n_c$	$c_{fact} = 0.02$	$c_{fact} = 0.06$	$c_{fact} = 0.08$	$c_{fact} = 0.09$	$c_{fact} = 0.099$
544/289/17	9/583	11/525	12/561	13/572	13/549
2112/1089/33	9/990	11/851	12/866	13/1039	13/901
8320/4225/65	9/1428	12/1523	13/1551	13/1426	14/1497
33024/16641/129	10/2687	12/2342	13/2309	14/2432	14/2380
74112/37249/193	10/3468	12/2843	13/2824	14/3282	15/3182

Table 4: $r_{tol} = 0.5$

$n_u/n_p/n_c$	$c_{fact} = 0.02$	$c_{fact} = 0.06$	$c_{fact} = 0.08$	$c_{fact} = 0.09$	$c_{fact} = 0.099$
544/289/17	9/531	11/501	12/511	12/497	13/556
2112/1089/33	9/991	12/979	13/1013	13/920	13/868
8320/4225/65	9/1538	12/1475	13/1459	14/1614	14/1547
33024/16641/129	10/2710	12/2217	13/2141	14/2424	14/2171
74112/37249/193	10/3507	13/3243	13/2743	14/2838	15/3111

Table 5: $r_{tol} = 0.9$

In Table 6 we can see what the influence of κ on calculations. For this experiment, we chose the CGM tolerance to be fixed $cgmtol = 10^{-12}$ simulating the exact solution of (5.2) and $tol = 10^{-8}$. We can see that the number of iterations as well as matrix-vector multiplications increases as the discretization steps are smaller. This result is expected and not very surprising. If we take a look at the number of iterations for different κ values, we can notice the increase in iterations as κ gets smaller and closer to 0.

$n_u/n_p/n_c$	$\kappa = 1$	$\kappa = 0.5$	$\kappa = 0.1$	$\kappa = 0.01$	$\kappa = 0.001$
544/289/17	4/988	4/1034	4/1133	4/1210	4/1177
2112/1089/33	4/1670	4/2213	5/2536	5/2912	5/3249
8320/4225/65	4/2575	5/3526	5/4079	6/5905	6/5836
33024/16641/129	3/3705	4/3891	6/7285	7/10511	7/10809
74112/37249/193	3/4576	4/6010	5/8763	7/13120	7/13901

Table 6: Influence of κ on the number of iterations (square domain)

In Figure 5 we can see the distribution of $\boldsymbol{\sigma}_\tau$ and scaled $\mathbf{u}_\tau$ along γ_C for different κ. The values of κ for each graph are as follows: top left $\kappa = 0.5$, top right $\kappa = 0.1$, bottom left $\kappa = 0.01$, bottom right $\kappa = 0.001$. It is easy to see how $\mathbf{u}_t$ increases when $|\boldsymbol{\sigma}_t| > \mathbf{g}$ and $\mathbf{u}_t = 0$ while $|\boldsymbol{\sigma}_t| \leq \mathbf{g}$ following conditions (2.6) - (2.7).

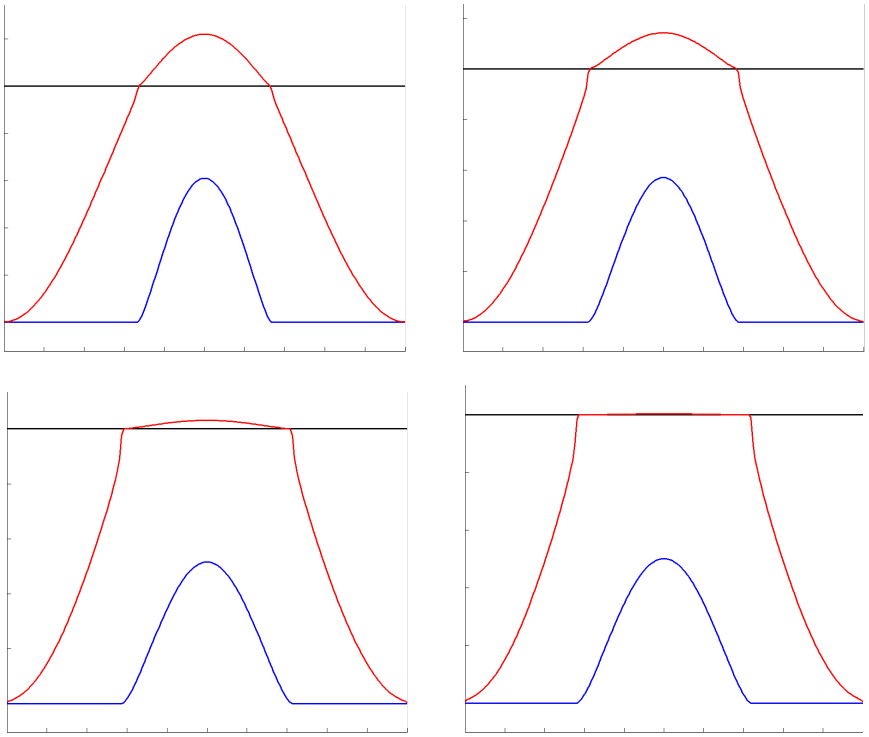

Figure 5: Distribution of $\boldsymbol{\sigma}_t$ (red) and scaled $\mathbf{u}_t$ (blue) along γ_C for different κ, the black line is the value of g (square domain)

Table 7 shows the accuracy of our results. The columns (4.7) - (4.10) represent relative accuracies of the respective optimality conditions. It was computed with $\kappa = 100$.

$n_u/n_p/n_c$	(4.7)	(4.8)	(4.9)	(4.10)	$iter/n_A$
544/289/17	$2.2972e-015$	$5.6447e-007$	$6.5393e-010$	$4.3955e-009$	8/291
2112/1089/33	$9.2627e-015$	$5.8831e-007$	$3.0119e-009$	$1.9463e-007$	8/442
8320/4225/65	$2.3752e-014$	$7.4602e-007$	$6.8951e-009$	$4.9144e-007$	8/569
33024/16641/129	$8.0755e-014$	$5.9838e-008$	$3.5579e-010$	$8.2878e-008$	9/1017
74112/37249/193	$1.6490e-013$	$1.1693e-007$	$4.7438e-010$	$9.1631e-008$	9/1169

Table 7: The accuracy of results (square domain)

6.2 Example 2

We consider the following domain: $\Omega = (0,5) \times (0,2) \setminus \bar{\mathcal{S}}$, $\mathcal{S} = (0,1) \times (0,1)$, $\nu = 1$, $\mathbf{f} = 0$, $\gamma_D = \gamma_{\text{top}} \cup \gamma_{\text{left}}$, $\mathbf{u}_{D|\gamma_{\text{top}}} = \mathbf{0}$, inflow is $\mathbf{u}_{D|\gamma_{\text{left}}} = (4(y-2)(1-y), 0)$, $\gamma_N = \gamma_{\text{right}}$, $\boldsymbol{\sigma}_N = \mathbf{0}$, $\gamma_S = \gamma_{\text{bottom}}$, and $g = 1$.

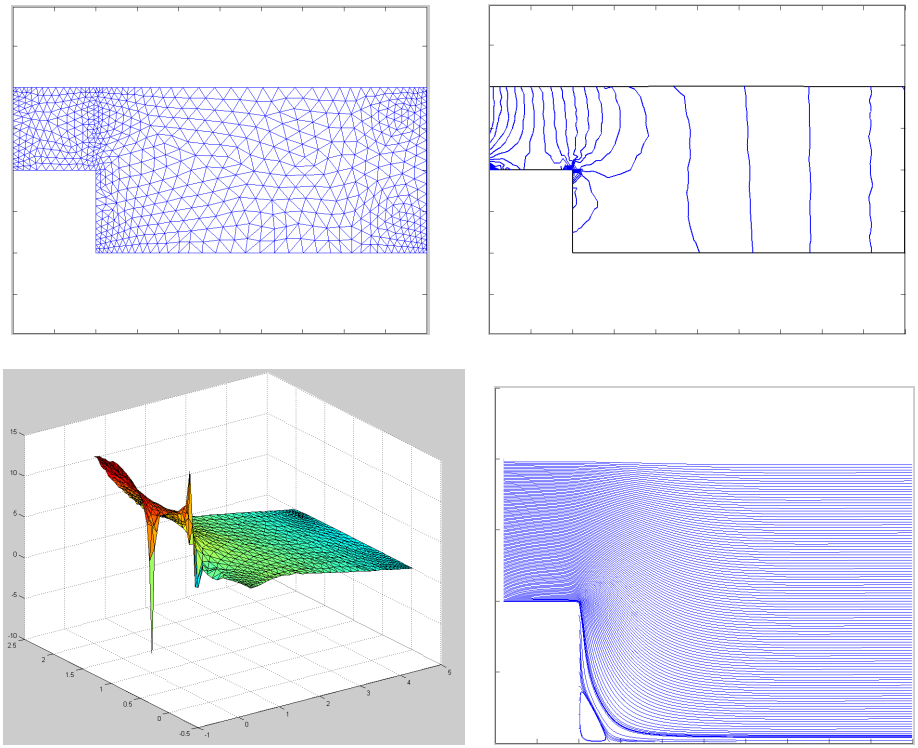

Figure 6: Mesh, isobars, pressure and velocity field (L-shaped domain)

The influence of κ on computations is shown in Table 8 for $tol = 10^{-4}$ and r_{tol}, c_{fact} from the previous example. The preconditioning effect is shown in Table 9. One can see that efficiency of computations is considerably higher. By * we denote situations, in which the maximal number of inner iterations was exceeded. We can still observe the increase in iterations with changing κ values as in the previous example, but this time it is not as consistent. This suggests that the

preconditioning works, saving a lot of computational time, while keeping the same accuracy of the results.

$n_u/n_p/n_c$	$\kappa = 1$	$\kappa = 0.5$	$\kappa = 0.1$	$\kappa = 0.01$	$\kappa = 0.001$
344/206/32	8/1313	8/1197	9/1755	10/2375	*
1352/744/64	8/2727	9/3776	9/4684	10/6559	11/8858
5366/2819/128	9/6539	9/7314	10/11801	11/19484	11/17000
21386/10965/256	9/13064	10/16384	10/23394	11/28977	*
85394/43241/512	10/23324	10/29940	10/36387	13/62109	*

Table 8: Influence of κ on the number of iterations (L-shaped domain)

$n_u/n_p/n_c$	$\kappa = 1$	$\kappa = 0.5$	$\kappa = 0.1$	$\kappa = 0.01$	$\kappa = 0.001$
344/206/32	7/108	8/140	8/113	8/113	10/143
1352/744/64	7/139	8/182	8/163	9/171	10/189
5366/2819/128	8/249	8/262	9/231	10/268	12/340
21386/10965/256	8/286	8/258	9/271	11/339	10/262
85394/43241/512	8/378	8/322	8/248	11/444	11/434

Table 9: Influence of κ on the number of iterations with preconditioning (L-shaped domain)

In Table 10 we show analogous tests computed by path-following interior point method (PF algorithm) [9, 11]. One can see that the computations performed by Algorithm ISSNM with preconditioning are more efficient.

$n_u/n_p/n_c$	$\kappa = 1$	$\kappa = 0.5$	$\kappa = 0.1$	$\kappa = 0.01$	$\kappa = 0.001$
344/206/32	18/347	18/351	19/397	18/381	18/387
1352/744/64	20/494	20/486	20/503	20/491	20/483
5366/2819/128	18/500	18/445	19/521	19/494	19/476
21386/10965/256	19/605	19/565	19/608	19/557	19/608
85394/43241/512	19/687	19/690	19/742	19/705	19/671

Table 10: Influence of κ on the number of iterations for the PF algorithm (L-shaped domain)

In the Table 11, we show the accuracy of computations for $\kappa = 10$ and $tol = 10^{-5}$.

$n_u/n_p/n_c$	(4.7)	(4.8)	(4.9)	(4.10)	$iter/n_A$
344/206/32	0.2995	$4.0941e-009$	$2.7461e-009$	0.0886	8/144
1352/744/64	0.1614	$3.4753e-008$	$3.2884e-008$	0.0511	7/143
5366/2819/128	0.0864	$1.5785e-008$	$2.6778e-008$	0.0284	7/181
21386/10965/256	0.0454	$9.5327e-010$	$1.7818e-009$	0.0148	8/280
85394/43241/512	0.0235	$5.5616e-010$	$9.1567e-010$	0.0073	8/392

Table 11: The accuracy of results for SSNM algorithm (L-shaped domain)

Figure 7 represents the distribution of $\boldsymbol{\sigma}_\tau$ and scaled $\mathbf{u}_\tau$ along γ_C for different κ. The values of κ for each graph are as follows: top left $\kappa = 1$, top right $\kappa = 0.5$, bottom left $\kappa = 0.1$, bottom right $\kappa = 0.01$.

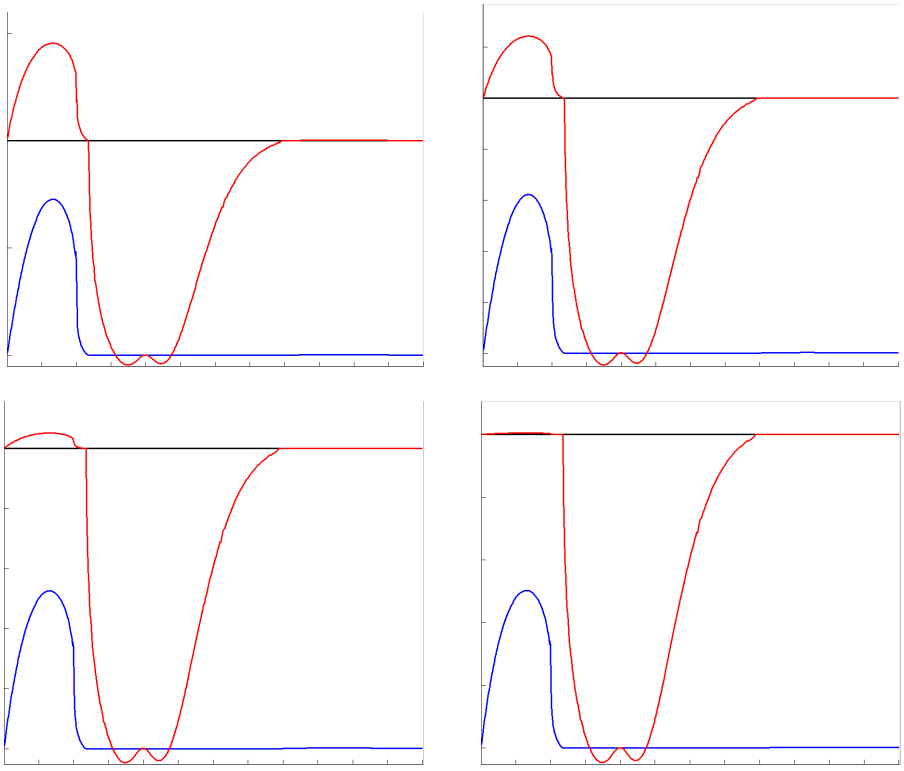

Figure 7: Distribution of $\boldsymbol{\sigma}_t$ (red) and scaled $\mathbf{u}_t$ (blue) along γ_C for different κ, the black line is the value of g (L-shaped domain)

7 Conclusion

The aim of the thesis was to propose an algorithm for numerical solution of the Stokes equations with monotonously increasing slip condition based on the semi-smooth Newton method. We have succesfully met this goal. Having the classical formulation of the problem, we have made the necessary changes and modifications to arrive at the suitable formulation of the problem, which could be used with the semi-smooth Newton method.

We have then succesfully implemented the algorithm in MATLAB environment, providing a detailed description and provided the effectiveness Providing a detailed description of the implementation in MATLAB environment We have implemented our suggested algorithm and made experiments on two types of domains. We have found out that as the adhesive coefficient κ got closer to 0, we needed the preconditioner for the conjugate gradient method, which allowed us to successfully finish our experiments. Furthermore, we showed that the SSNM algorithm is more efficient than the path-following variant of the interior-point method suggested in [11].

The proposed algorithm is a promising method for solving more sophisticated flow problems, i.e. problem with the convective term, time dependent problem, etc. Also an extension to 3D case could be in principle possible.

Jan Pacholek

References

[1] J. Koko,
Vectorized Matlab codes for the Stokes Problem with P1-bubble/P1 Finite Element, at: `http://www.isima.fr/~jkoko/Codes/StokesP1BubbleP1.pdf`.

[2] D. Arnold, F. Brezzi, M. Fortin,
A stable sinite element for the Stokes equations, Calcolo, 337-344 21, 1984.

[3] Howard C. Elman, David J. Silvester, Andrew J. Wathen,
Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, OUP Oxford, 19.5.2005.

[4] J. Haslinger, I. Hlaváček, J. Nečas,
Numerical methods for for unilateral problems in solid mechanics, Handbook of Numerical Analysis, vol. IV, Ciarlet, P.G. and Lions, J.L. eds., North-Holland, Amsterdam, 313-485, 1996.

[5] J. Nocedal, S. J. Wright,
Numerical Optimization, Springer-Verlag, New York, 1999.

[6] X. Chen, Z. Nashed, L. Qi,
Smoothing methods and semismooth methods for non-differentiable operator equations, SIAM J. Numer. Anal. **38**, 1200–1216, 2000.

[7] G. H. Golub, C. F. Van Loan,
Matrix computation, The Johns Hopkins University Press: Baltimore, 1996.

[8] M. Hintermüller, K. Ito, K. Kunisch,
The primal-dual active set strategy as a semismooth Newton method, SIAM J. Optim. **13**, 865–888, 2003.

[9] R. Kučera, M. Netuka, J. Machalová, P. Ženčák,
An interior point algorithm for the minimization arising from the 3d contact problems with friction, Optimization Methods and Software, **28**, 6, 1195-1217, 2013.

[10] H. Fujita,
A mathematical analysis of motions of viscous incompressible fluid under leak or slip boundary conditions, Sūrikaisekikenkyūsho Kōkyūroku (888), 199-216, 1994.

[11] R. Kučera, J.Haslinger, V.Šátek, M. Jarošová,
Efficient methods for solving the Stokes problem with slip boundary conditions, Accepted in Mathematics and Computers in Simulation, 2016.

[12] Navier,
C.L.M.H., Mem.Acad.R.Sci.Inst., France, I 414, 1823.

A Green's theorem

Theorem A.1

$$
\begin{aligned}
&1)\ \int_\Omega \frac{\partial u}{\partial x_i} v = -\int_\Omega u \frac{\partial v}{\partial x_i} + \int_{\partial\Omega} uvn_i, && u, v \in H^1(\Omega) \\
&2)\ \int_\Omega \frac{\partial^2 u}{\partial x_i^2} v = -\int_\Omega \frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_i} + \int_{\partial\Omega} \frac{\partial u}{\partial x_i} vn_i, && u \in H^2(\Omega), v \in H^1(\Omega) \\
&3)\ \int_\Omega \triangle uv = -\int_\Omega \nabla u \cdot \nabla v + \int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}} v, && u \in H^2(\Omega), v \in H^1(\Omega)
\end{aligned}
\tag{A.1}
$$

n_i is the i-th element of the normal outward vector n to $\partial\Omega$